矩阵计算讲义

潘建瑜
jypan@math.ecnu.edu.cn
华东师范大学数学系

May 26, 2014

[0]本讲义仅供课堂使用

目 录

EAST CHINA NORMAL UNIVERSITY

算 法 目 录

EAST CHINA NORMAL UNIVERSITY
華東師範大學

第一章　引言

本课程学习内容

- 线性方程组的直接解法 (Gauss 消去法 / LU 分解)
- 线性最小二乘问题的数值算法
- 非对称矩阵的特征值计算
- 对称矩阵特征值计算与奇异值分解
- 线性方程组迭代算法
- 特征值问题的迭代算法

主要参考资料

关于矩阵计算的参考资料有

- G. Golub and C. F. van Loan, “Matrix Computations (3rd),” Johns Hopkins University Press, 1996.
  对应中文版为:《矩阵计算》(第三版), 袁亚湘等译, 科学出版社, 2001.
- J. W. Demmel, “Applied Numerical Linear Algebra,” SIAM, 1997.
  对应中文版为:《应用数值线性代数》, 王国荣译, 人民邮电出版社, 2007.
- L. N. Trefethen and D. Bau, III, “Numerical Linear Algebra,” SIAM, 1997.
  对应中文版为:《数值线性代数》, 陆金甫等译, 人民邮电出版社, 2006.
- 徐树方, “矩阵计算的理论与方法,” 北京大学出版社, 1995.
- 曹志浩, “数值线性代数,” 复旦大学出版社, 1996.

课程主页

http://math.ecnu.edu.cn/~jypan/Teaching/MatrixComp/index.html

EAST CHINA NORMAL UNIVERSITY

1.1 数值线性代数基本问题

1.1.1 计算数学介绍

计算数学, 也称数值分析或计算方法, 其定义可以参考 Golub 教授的 History of numerical linear algebra: A personal view [14], 或 Trefethen 教授的 Numerical analysis [40]. 一般来说, 计算数学主要研究的是如何计算数学问题的近似解或数值解, 包括算法的收敛性和稳定性, 以及解的误差估计.

从事现代科学研究的方法有三种: 实验, 理论和科学计算. 科学计算就是通过数学建模将实际问题转化为数学问题, 然后数学问题进行离散和数值求解, 从而得到原问题的近似解. 随着计算机的高速发展, 科学计算在解决现代科学技术问题中所起的作用越来越大, 并已经渗透到科学技术的各个领域. 计算数学是科学计算的基础, 其研究内容包括函数逼近 (代数插值与最佳逼近), 数值积分与数值微分, 数值线性代数 (线性方程组求解, 特征值与特征向量等), 非线性方程 (组) 数值解法, 常微分方程数值解法, 偏微分方程数值解法等.

国家自然科学基金委员会关于计算数学的分类 (2012):

数学 (A01)

- 计算数学与科学工程计算 (A0117)
    - 偏微分方程数值解 (A011701)
    - 流体力学中的数值计算 (A011702)
    - 一般反问题的计算方法 (A011703)
    - 常微分方程数值计算 (A011704)
    - 数值代数 (A011705)
    - 数值逼近与计算几何 (A011706)
    - 谱方法及高精度数值方法 (A011707)
    - 有限元和边界元方法 (A011708)
    - 多重网格技术与区域分解 (A011709)
    - 自适应方法 (A011720)
    - 并行计算 (A011711)
- 运筹学

数值分析的任务是

- 设计求解各种实际问题的高效可靠的数值方法;
- 对求得的数值解的精度进行评估;
- 研究数值算法在计算机上的实现.

一个好的数值方法一般需满足以下几点:

- 易于在计算机上实现;
- 有可靠的理论分析, 即收敛性稳定性等有数学理论保证;
- 有良好的计算复杂性, 即尽可能地节省计算时间和存储空间;
- 要有具体的数值试验来证明是行之有效的.

1.1.2 数值线性代数的研究内容

- 数值代数, 包含数值线性代数和数值非线性代数.
- 数值线性代数, 也称矩阵计算, 主要研究以下问题:

- 线性方程组求解
$$Ax = b, \quad A \in \mathbb{R}^{n\times n} \text{ 非奇异}$$
- 最小二乘问题
$$\min_{x\in\mathbb{R}^n} \|Ax - b\|_2, \quad A \in \mathbb{R}^{m\times n},\ m \geq n$$
- 矩阵特征值问题
$$Ax = \lambda x, \quad A \in \mathbb{R}^{n\times n},\ x \neq 0,\ \lambda \in \mathbb{C}$$
- 矩阵奇异值问题
$$A^T Ax = \sigma^2 x, \quad A \in \mathbb{R}^{m\times n},\ x \neq 0,\ \sigma \geq 0$$
- 其它: 广义特征值问题, 非线性特征值问题, 矩阵方程, 特征值反问题, 张量计算, $\cdots\cdots$

- 主要任务
  - 算法设计: 构造计算近似解的方法
  - 算法分析: 误差估计、收敛性、稳定性、计算复杂性、计算精度等
  - 算法实现: 编程实现、软件开发
- 基本研究方法
  - 矩阵分解
  - 扰动分析
  - 舍入误差对算法的影响
  - 算法的收敛速度分析
  - 工业化的软件实现

☞ 问题的特殊结构对算法的设计具有非常重要的影响.

1.1.3 一些基本记号

- 实部与虚部: $\mathrm{Re}(z)$ 表示复数 z 的实部, $\mathrm{Im}(z)$ 表示复数 z 的虚部;
- 矩阵: 通常用大写字母表示, 如 A, B, X ;
- 向量: 小写字母, 如 x, y, z ;
- 标量: 小写字母或希腊字母, 如 a, b, c, α, β ;
- 转置: x^T 和 A^T 表示向量和矩阵的普通转置, x^* 和 A^* 表示向量和矩阵的共轭转置;
- 下标: 用 a_{ij}, $A(i,j)$ 或 $(A+B)_{i,j}$ 表示矩阵的元素;
- 冒号的作用: (取自 Matlab)
  - $a\!:\!h\!:\!b \to$ 生成一个等差序列: a 为首项, h 为公差, 最后一项 $\leq b$
    当 $h = 1$ 时, 可简写成 $a : b$
  - $x(2\!:\!5) \to [x(2), x(3), x(4), x(5)]$
  - $A(1\!:\!3, 2\!:\!5) \to A$ 的第 1 至第 3 行与第 2 至第 5 列组成的子矩阵
  - $A(1\!:\!3, :) \to A$ 的第 1 至第 3 行组成的子矩阵

1.2 线性代数基础

1.2.1 线性空间

线性空间是线性代数最基本的概念之一, 它是定义在某个数域上并满足一定条件的一个集合. 我们首先给出数域的概念.

定义 1.1 (数域) 设 $\mathbb{F}$ 是包含 *0* 和 *1* 的一个数集, 如果 $\mathbb{F}$ 中的任意两个数的和, 差, 积, 商 (除数不为 *0*) 仍然在 $\mathbb{F}$ 中, 则称 $\mathbb{F}$ 为一个**数域**.

常见的数域有: 有理数域 $\mathbb{Q}$, 实数域 $\mathbb{R}$ 和复数域 $\mathbb{C}$.

定义 1.2 (线性空间) 设 $\mathcal{S}$ 是一个非空集合, $\mathbb{F}$ 是一个数域. 在 $\mathcal{S}$ 上定义一种代数运算, 称为**加法**, 记为 "+" (即对任意 $\alpha, \beta \in \mathcal{S}$, 都存在唯一的 $\gamma \in \mathcal{S}$, 使得 $\gamma = \alpha + \beta$), 满足

(1) **交换律**: $\alpha + \beta = \beta + \alpha, \quad \forall\, \alpha, \beta \in \mathcal{S}$;

(2) **结合律**: $(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma), \quad \forall\, \alpha, \beta, \gamma \in \mathcal{S}$;

(3) **零元素**: 存在一个元素 *0*, 使得 $\alpha + 0 = \alpha, \quad \forall\, \alpha \in \mathcal{S}$;

(4) **逆运算**: 对任意 $\alpha \in \mathcal{S}$, 都存在**负元素** $\beta \in \mathcal{S}$, 使得 $\alpha + \beta = 0$, 记 $\beta = -\alpha$;

定义一个从 $\mathbb{F} \times \mathcal{S}$ 到 $\mathcal{S}$ 的代数运算, 称为**数乘**, 记为 "$\cdot$" (即对任意 $k \in \mathbb{F}$ 和任意 $\alpha \in \mathcal{S}$, 都存在唯一的 $\beta \in \mathcal{S}$, 使得 $\beta = k \cdot \alpha$), 满足

(1) $1 \cdot \alpha = \alpha, \quad 1 \in \mathbb{F}, \forall\, \alpha \in \mathcal{S}$;

(2) $k \cdot (l \cdot \alpha) = (kl) \cdot \alpha, \quad \forall\, k, l \in \mathbb{F},\ \alpha \in \mathcal{S}$;

(3) $(k + l) \cdot \alpha = k \cdot \alpha + l \cdot \alpha, \quad \forall\, k, l \in \mathbb{F},\ \alpha \in \mathcal{S}$;

(4) $k \cdot (\alpha + \beta) = k \cdot \alpha + k \cdot \beta, \quad \forall\, k \in \mathbb{F},\ \alpha, \beta \in \mathcal{S}$;

则称 $\mathcal{S}$ 是数域 $\mathbb{F}$ 上的一个**线性空间**. 为了表示方便, 通常省略数乘符号, 即将 $k \cdot \alpha$ 写成 $k\alpha$.

例 1.1 常见的线性空间有:

- $\mathbb{R}^n \to$ 所有 n 维实向量组成的集合, 是 $\mathbb{R}$ 上的线性空间;
- $\mathbb{C}^n \to$ 所有 n 维复向量组成的集合, 是 $\mathbb{C}$ 上的线性空间;
- $\mathbb{R}^{m\times n} \to$ 所有 $m \times n$ 阶实矩阵组成的集合, 是 $\mathbb{R}$ 上的线性空间;
- $\mathbb{C}^{m\times n} \to$ 所有 $m \times n$ 阶复矩阵组成的集合, 是 $\mathbb{C}$ 上的线性空间.

定义 1.3 (线性子空间) 设 $\mathcal{S}$ 是一个线性空间, $\mathcal{W}$ 是 $\mathcal{S}$ 的一个非空子集合. 如果 $\mathcal{W}$ 关于 $\mathcal{S}$ 上的加法和数乘也构成一个线性空间, 则称 $\mathcal{W}$ 为 $\mathcal{S}$ 的一个**线性子空间**, 有时简称**子空间**.

定理 1.1 设 $\mathcal{S}$ 是数域 $\mathbb{F}$ 上的一个线性空间, $\mathcal{W}$ 是 $\mathcal{S}$ 的一个非空子集合. 则 $\mathcal{W}$ 是 $\mathcal{S}$ 的一个子空间的充要条件是 $\mathcal{W}$ 关于加法和数乘封闭, 即

(1) 对任意 $\alpha, \beta \in \mathcal{W}$, 有 $\alpha + \beta \in \mathcal{W}$;

(2) 对任意 $k \in \mathbb{F}$ 和任意 $\alpha \in \mathcal{W}$, 有 $k\alpha \in \mathcal{W}$.

设 $A \in \mathbb{C}^{m\times n}$, 则 A 可以看作是从 $\mathbb{C}^n$ 到 $\mathbb{C}^m$ 的一个线性变换 (或线性映射). 我们分别称

$$\mathrm{Ker}(A) \triangleq \left\{ x \in \mathbb{C}^n \ :\ Ax = 0 \right\} \subseteq \mathbb{C}^n$$

和

$$\mathrm{Ran}(A) \triangleq \{\, y \in \mathbb{C}^m \,:\, y = Ax,\ x \in \mathbb{C}^n \,\} \subseteq \mathbb{C}^m$$

为 A 的零空间(核) 和像空间(列空间, 值域). 可以证明, $\mathrm{Ker}(A)$ 是 $\mathbb{C}^n$ 的线性子空间, $\mathrm{Ran}(A)$ 是 $\mathbb{C}^m$ 的线性子空间.

设 $x_1, x_2, \ldots, x_k \in \mathbb{C}^n$, 记

$$\mathrm{span}(x_1, x_2, \ldots, x_k) \triangleq \{\, \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k \,:\, \alpha_1, \alpha_2, \ldots, \alpha_k \in \mathbb{C} \,\},$$

则 $\mathrm{span}(x_1, x_2, \ldots, x_k)$ 构成 $\mathbb{C}^n$ 的一个线性子空间, 称为由 $x_1, x_2, \ldots, x_k$ 张成的子空间. 特别地, 记 $\mathrm{span}(A)$ 为由 A 的所有列向量张成的子空间. 易知

$$\mathrm{Ran}(A) = \mathrm{span}(A).$$

记 $\dim(\mathcal{S})$ 为线性空间 $\mathcal{S}$ 的维数, 设 $A \in \mathbb{C}^{m\times n}$, 则有

- $\dim(\mathrm{Ker}(A)) = n - \mathrm{rank}(A)$
- $\dim(\mathrm{Ran}(A)) = \mathrm{rank}(A)$
- $\dim(\mathrm{Ker}(A)) + \dim(\mathrm{Ran}(A)) = n$
- $\dim(\mathrm{Ran}(A^T)) = \dim(\mathrm{Ran}(A^T A))$

定理 1.2 (维数公式) 设 $\mathcal{S}_1, \mathcal{S}_2$ 是线性空间 $\mathcal{S}$ 的两个有限维线性子空间, 则 $\mathcal{S}_1 + \mathcal{S}_2$ 和 $\mathcal{S}_1 \cap \mathcal{S}_2$ 也都是 $\mathcal{S}$ 的线性子空间, 且

$$\dim(\mathcal{S}_1 + \mathcal{S}_2) + \dim(\mathcal{S}_1 \cap \mathcal{S}_2) = \dim(\mathcal{S}_1) + \dim(\mathcal{S}_2).$$

定义 1.4 (直和) 设 $\mathcal{S}_1, \mathcal{S}_2$ 是线性空间 $\mathcal{S}$ 的两个线性子空间, 如果 $\mathcal{S}_1 + \mathcal{S}_2$ 中的任一元素 x 都可以唯一表示成

$$x = x_1 + x_2, \quad x_1 \in \mathcal{S}_1, x_2 \in \mathcal{S}_2,$$

则称 $\mathcal{S}_1 + \mathcal{S}_2$ 为直和, 记为 $\mathcal{S}_1 \oplus \mathcal{S}_2$.

定理 1.3 设 $\mathcal{S}_1, \mathcal{S}_2$ 是线性空间 $\mathcal{S}$ 的两个线性子空间, 则下面的论述等价

(1) $\mathcal{S}_1 + \mathcal{S}_2$ 是直和;

(2) $\mathcal{S}_1 \cap \mathcal{S}_2 = \{0\}$;

(3) $\dim(\mathcal{S}_1) + \dim(\mathcal{S}_2) = \dim(\mathcal{S}_1 + \mathcal{S}_2)$;

(4) $\mathcal{S}_1 + \mathcal{S}_2$ 中的零元素表示方法唯一, 即若 $0 = x_1 + x_2, x_1 \in \mathcal{S}_1, x_2 \in \mathcal{S}_2$, 则 $x_1 = x_2 = 0$.

定理 1.4 设 $\mathcal{S}_1$ 是线性空间 $\mathcal{S}$ 的一个线性子空间, 则存在 $\mathcal{S}$ 的另一个线性子空间 $\mathcal{S}_2$, 使得 $\mathcal{S} = \mathcal{S}_1 \oplus \mathcal{S}_2$.

1.2.2 内积空间

定义 1.5 (内积空间) 设 $\mathcal{S}$ 是数域 $\mathbb{F}$ 上的一个线性空间, 定义一个从 $\mathcal{S}\times\mathcal{S}$ 到 $\mathbb{F}$ 的代数运算, 记为 "$(\cdot,\cdot)$", 即对任意 $x,y\in\mathcal{S}$, 都存在唯一的 $f\in\mathbb{F}$, 使得 $f=(x,y)$, 如果满足

(1) $(x,y)=\overline{(y,x)}, \quad \forall\, x,y\in\mathcal{S}$;

(2) $(x+y,z)=(x,y)+(x,z), \quad \forall\, x,y,z\in\mathcal{S}$;

(3) $(kx,y)=k(x,y), \quad \forall\, k\in\mathbb{F},\ x,y\in\mathcal{S}$;

(4) $(x,x)\geq 0$, 等号当且仅当 $x=0$ 时成立;

则称 $(\cdot,\cdot)$ 为 $\mathcal{S}$ 上的一个内积, 定义了内积的线性空间称为内积空间.

特别地, 我们称定义在实数域 $\mathbb{R}$ 上的内积空间为 Euclidean 空间 (或欧氏空间), 定义在复数域 $\mathbb{C}$ 上的内积空间为 酉空间.

例 1.2 在线性空间 $\mathbb{C}^n$ 上定义内积

$$(x,y)=y^*x=\sum_{i=1}^{n}x_i\bar{y}_i,$$

则 $\mathbb{C}^n$ 构成一个内积空间.

例 1.3 对任意 $A,B\in\mathbb{R}^{m\times n}$, 定义

$$(A,B)=\mathrm{tr}(B^TA),$$

其中 $\mathrm{tr}(\cdot)$ 表示矩阵的迹, 即对角线元素之和, 则可以证明 (A,B) 是一个内积, 因此 $\mathbb{R}^{m\times n}$ 构成一个欧氏空间.

定义 1.6 (正交) 设 $\mathcal{S}$ 是内积空间, $x,y\in\mathcal{S}$, 如果 $(x,y)=0$, 则称 x 与 y 正交, 记为 $x\perp y$; 设 $\mathcal{S}_1$ 是 $\mathcal{S}$ 的子空间, $x\in\mathcal{S}$, 如果对任意 $y\in\mathcal{S}_1$ 都有 $(x,y)=0$, 则称 x 与 $\mathcal{S}_1$ 正交, 记为 $x\perp\mathcal{S}_1$; 设 $\mathcal{S}_1$, $\mathcal{S}_2$ 是 $\mathcal{S}$ 的两个子空间, 如果对任意 $x\in\mathcal{S}_1$, 都有 $x\perp\mathcal{S}_2$, 则称 $\mathcal{S}_1$ 与 $\mathcal{S}_2$ 正交, 记为 $\mathcal{S}_1\perp\mathcal{S}_2$.

定理 1.5 设 $\mathcal{S}_1,\mathcal{S}_2$ 是内积空间 $\mathcal{S}$ 的两个子空间, 如果 $\mathcal{S}_1\perp\mathcal{S}_2$, 则 $\mathcal{S}_1+\mathcal{S}_2$ 是直和.

定义 1.7 (正交补) 设 $\mathcal{S}_1$ 是内积空间 $\mathcal{S}$ 的一个子空间, 则 $\mathcal{S}_1$ 的正交补定义为

$$\mathcal{S}_1^{\perp}\triangleq\{\,x\in\mathcal{S}\ :\ x\perp\mathcal{S}_1\,\}.$$

定理 1.6 设 $\mathcal{S}_1$ 是内积空间 $\mathcal{S}$ 的一个有限维子空间, 则 $\mathcal{S}_1^{\perp}$ 存在唯一, 且

$$\mathcal{S}=\mathcal{S}_1\oplus\mathcal{S}_1^{\perp}.$$

1.2.3 向量范数

定义 1.8 (向量范数) 若函数 $f:\mathbb{C}^n\to\mathbb{C}$ 满足

(1) $f(x)\geq 0, \forall\, x\in\mathbb{C}^n$ 且等号当且仅当 $x=0$ 时成立;

(2) $f(\alpha x)=|\alpha|\cdot f(x), \forall\, x\in\mathbb{C}^n, \alpha\in\mathbb{C}$;

(3) $f(x+y)\leq f(x)+f(y), \forall x,y\in\mathbb{C}^n$;

则称 $f(x)$ 为 $\mathbb{C}^n$ 上的范数, 通常记作 $\|\cdot\|$.

☞ 相类似地, 我们可以定义实数空间 $\mathbb{R}^n$ 上的向量范数.

例 1.4 常见的向量范数:

- 1-范数: $\|x\|_1=|x_1|+|x_2|+\cdots+|x_n|$;
- 2-范数: $\|x\|_2=\sqrt{|x_1|^2+|x_2|^2+\cdots+|x_n|^2}$;
- ∞-范数: $\|x\|_\infty=\max\limits_{1\leq i\leq n}|x_i|$;
- p-范数: $\|x\|_p=\left(\sum\limits_{i=1}^n|x_i|^p\right)^{1/p}, \quad 1\leq p<\infty$.

定义 1.9 (范数的等价性) 设 $\|\cdot\|_\alpha$ 与 $\|\cdot\|_\beta$ 是 $\mathbb{C}^n$ 空间上的两个向量范数, 若存在正常数 c_1, c_2, 使得

$$c_1\|x\|_\alpha\leq\|x\|_\beta\leq c_2\|x\|_\alpha$$

对任意 $x\in\mathbb{C}^n$ 都成立, 则称 $\|\cdot\|_\alpha$ 与 $\|\cdot\|_\beta$ 是等价的.

定理 1.7 $\mathbb{C}^n$ 空间上的所有向量范数都是等价的, 特别地, 有

$$\begin{aligned}
&\|x\|_2\leq\|x\|_1\leq\sqrt{n}\,\|x\|_2,\\
&\|x\|_\infty\leq\|x\|_2\leq\sqrt{n}\,\|x\|_\infty,\\
&\|x\|_\infty\leq\|x\|_1\leq n\,\|x\|_\infty.
\end{aligned}$$

☞ 有限维赋范线性空间上的所有范数都是等价的.

定理 1.8 (Cauchy-Schwartz 不等式) 设 $(\cdot,\cdot)$ 是 $\mathbb{C}^n$ 上的内积, 则对任意 $x,y\in\mathbb{C}^n$, 有

$$|(x,y)|^2\leq(x,x)\cdot(y,y).$$

推论 1.9 设 $(\cdot,\cdot)$ 是 $\mathbb{C}^n$ 上的内积, 则 $\|x\|\triangleq\sqrt{(x,x)}$ 是 $\mathbb{C}^n$ 上的一个向量范数.

定理 1.10 (范数的连续性) 设 $\|\cdot\|$ 是 $\mathbb{C}^n$ 上的一个向量范数, 则 $f(x)\triangleq\|x\|$ 是 C^n 上的连续函数.

定义 1.10 (向量序列的收敛) 设 $\{x^{(k)}\}_{k=1}^{\infty}$ 是 C^n 中的一个向量序列, 如果存在 $x \in \mathbb{C}^n$, 使得

$$\lim_{k\to\infty} x_i^{(k)} = x_i, \quad i = 1, 2, \ldots, n,$$

则称 $\{x^{(k)}\}$ (按分量) 收敛到 x, 记为 $\lim\limits_{k\to\infty} x^{(k)} = x$.

定理 1.11 设 $\|\cdot\|$ 是 $\mathbb{C}^n$ 上的任意一个向量范数, 则 $\lim\limits_{k\to\infty} x^{(k)} = x$ 的充要条件是

$$\lim_{k\to\infty} \|x^{(k)} - x\| = 0.$$

1.2.4 矩阵范数

定义 1.11 (矩阵范数) 若函数 $f : \mathbb{C}^{m\times n} \to \mathbb{C}$ 满足

(1) $f(A) \geq 0, \forall\, A \in \mathbb{C}^{m\times n}$ 且等号当且仅当 $A = 0$ 时成立;

(2) $f(\alpha A) = |\alpha| \cdot f(A), \forall\, A \in \mathbb{C}^{m\times n}, \alpha \in \mathbb{C}$;

(3) $f(A + B) \leq f(A) + f(B), \forall A, B \in \mathbb{C}^{m\times n}$;

则称 $f(x)$ 为 $\mathbb{C}^{m\times n}$ 上的范数, 通常记作 $\|\cdot\|$.

设 $\|\cdot\|$ 是 $\mathbb{C}^{m\times n}$ 上的范数, 若对任意 $A \in \mathbb{C}^{m\times n}$ 和任意 $x \in \mathbb{C}^n$, 有

$$\|Ax\| \leq \|A\| \cdot \|x\|, \tag{1.1}$$

则称矩阵范数 $\|\cdot\|$ 与向量范数**相容**, 这里的 $\|Ax\|$ 和 $\|x\|$ 分别为 C^m 和 C^n 上的向量范数.

☞ 定义中的 $\mathbb{C}^{n\times n}$ 可换成 $\mathbb{C}^{m\times n}$.

☞ 类似地, 我们可以定义 $\mathbb{R}^{m\times n}$ 上的矩阵范数.

☞ 在不加特别指出时, 本讲义中所讨论的矩阵范数都是指相容范数.

例 1.5 常见的矩阵范数:

- F-范数

$$\|A\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2}\,;$$

- p-范数

$$\|A\|_p = \sup_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p}\,.$$

引理 1.1 (算子范数, 诱导范数, 导出范数) 设 $\|\cdot\|$ 是 $\mathbb{R}^n$ 上的向量范数, 则

$$\|A\| \triangleq \sup_{x\in\mathbb{R}^n,\, x\neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|$$

是 $\mathbb{R}^{n\times n}$ 上的范数, 称为算子范数, 或诱导范数, 导出范数.

引理 1.2 可以证明:

(1) 1-范数 (列范数): $\|A\|_1 = \max\limits_{1\le j\le n}\left(\sum\limits_{i=1}^{n}|a_{ij}|\right)$;

(2) ∞-范数 (行范数): $\|A\|_\infty = \max\limits_{1\le i\le n}\left(\sum\limits_{j=1}^{n}|a_{ij}|\right)$;

(3) 2-范数: $\|A\|_2 = \sqrt{\rho(A^TA)}$.

定理 1.12 (矩阵范数的等价性) $\mathbb{R}^{n\times n}$ 空间上的所有范数都是等价的, 特别地, 有

$$\frac{1}{\sqrt{n}}\|A\|_2 \le \|A\|_1 \le \sqrt{n}\,\|A\|_2,$$
$$\frac{1}{\sqrt{n}}\|A\|_2 \le \|A\|_\infty \le \sqrt{n}\,\|A\|_2,$$
$$\frac{1}{n}\|A\|_\infty \le \|A\|_1 \le n\,\|A\|_\infty,$$
$$\frac{1}{\sqrt{n}}\|A\|_1 \le \|A\|_F \le \sqrt{n}\,\|A\|_2.$$

性质 1.1 范数的性质:

(1) 对任意相容范数 $\|\cdot\|$, 有 $\|A^k\| \le \|A\|^k$;

(2) 对任意算子范数 $\|\cdot\|$, 有 $\|Ax\| \le \|A\|\cdot\|x\|$, $\|AB\| \le \|A\|\cdot\|B\|$, 即算子范数是相容范数;

(3) $\|Ax\|_2 \le \|A\|_F\cdot\|x\|_2$, $\|AB\|_F \le \|A\|_F\cdot\|B\|_F$, 即 F-范数是相容范数;

(4) F-范数不是算子范数;

(5) $\|\cdot\|_2$ 和 $\|\cdot\|_F$ 是酉不变范数, 即对任意酉矩阵 U, V, 有

$$\|UA\|_2 = \|AV\|_2 = \|UAV\|_2 = \|A\|_2\,,$$
$$\|UA\|_F = \|AV\|_F = \|UAV\|_F = \|A\|_F$$

(6) $\|A^T\|_2 = \|A\|_2$, $\|A^T\|_1 = \|A\|_\infty$;

(7) 若 A 是正规矩阵, 则 $\|A\|_2 = \rho(A)$.

1.2.5 投影变换

设 $\mathcal{S}_1$ 和 $\mathcal{S}_2$ 是内积空间 $\mathcal{S}$ 的两个子空间, 且 $\mathcal{S} = \mathcal{S}_1 \oplus \mathcal{S}_2$. 则 $\mathcal{S}$ 中的任意向量 x 都可唯一表示为

$$x = x_1 + x_2, \quad x_1 \in \mathcal{S}_1,\ x_2 \in \mathcal{S}_2.$$

我们称 x_1 为 x 沿 $\mathcal{S}_2$ 到 $\mathcal{S}_1$ 上的 投影, 记为 $x|_{\mathcal{S}_1}$.

设线性变换 $P : \mathcal{S} \to \mathcal{S}$. 如果对任意 $x \in \mathcal{S}$, 都有

$$Px = x|_{\mathcal{S}_1},$$

则称 P 是从 $\mathcal{S}$ 沿子空间 $\mathcal{S}_2$ 到子空间 $\mathcal{S}_1$ 上的 **投影变换** (也称 **投影算子** 或 **投影矩阵**).

设 P 是投影变换, 容易验证, $\mathcal{S}_1 = \mathrm{Ran}(P), \mathcal{S}_2 = \mathrm{Ker}(P)$, 即存在直和分解

$$\mathcal{S} = \mathrm{Ran}(P) \oplus \mathrm{Ker}(P).$$

定理 1.13 矩阵 $P \in \mathbb{R}^{n\times n}$ 是投影矩阵的充要条件是 P 是幂等矩阵, 即 $P^2 = P$.

性质 1.2 设 $P \in \mathbb{R}^{n\times n}$ 是一个投影矩阵, 则

(1) $I - P$ 也是一个投影矩阵, 且 $\mathrm{Ker}(P) = \mathrm{Ran}(I - P)$;

(2) P^T 也是一个投影矩阵.

性质 1.3 设 $P \in \mathbb{R}^{n\times n}$ 是一个投影矩阵, 则

$$\mathbb{R}^n = \mathrm{Ran}(P) \oplus \mathrm{Ker}(P).$$

反之, 若 $\mathcal{S}_1$ 和 $\mathcal{S}_2$ 是 $\mathbb{R}^n$ 的两个子空间, 且 $\mathbb{R}^n = \mathcal{S}_1 \oplus \mathcal{S}_2$, 存在唯一的投影矩阵 P, 使得

$$\mathrm{Ran}(P) = \mathcal{S}_1, \quad \mathrm{Ker}(P) = \mathcal{S}_2,$$

即对任意向量 $x \in \mathbb{R}^n$, 有

$$Px \in \mathcal{S}_1, \quad x - Px \in \mathcal{S}_2.$$

由上面的性质可知, 投影矩阵由其像空间和零空间唯一确定.

性质 1.4 设 $\mathcal{S}_1$ 和 $\mathcal{S}_2$ 是 $\mathbb{R}^n$ 的两个子空间, 且维数相同. 如果 $\mathcal{S}_1 \cap \mathcal{S}_2^{\perp} = \{0\}$ (或 $\mathbb{R}^n = \mathcal{S}_1 \oplus \mathcal{S}_2^{\perp}$), 则存在唯一的投影矩阵 P, 使得

$$\mathrm{Ran}(P) = \mathcal{S}_1, \quad \mathrm{Ker}(P) = \mathcal{S}_2^{\perp},$$

即 P 是 $\mathcal{S}_1$ 上与 $\mathcal{S}_2$ 正交的投影矩阵. 另外,

$$Px = 0 \quad \text{当且仅当} \quad x \perp \mathcal{S}_2.$$

设 $\mathcal{S}_1$ 和 $\mathcal{S}_2$ 是 $\mathbb{R}^n$ 的两个 m 维子空间, 且 $\mathbb{R}^n = \mathcal{S}_1 \oplus \mathcal{S}_2^{\perp}$. 令 $v_1, v_2, \ldots, v_m$ 和 $w_1, w_2, \ldots, w_m$ 分别是 $\mathcal{S}_1$ 和 $\mathcal{S}_2$ 的一组基.

性质 1.5 设 $P \in \mathbb{R}^{n\times n}$ 是 $\mathcal{S}_1$ 上与 $\mathcal{S}_2$ 正交的投影矩阵, 则

$$P = V(W^T V)^{-1} W^T, \tag{1.2}$$

其中 $V = [v_1, v_2, \ldots, v_m]$, $W = [w_1, w_2, \ldots, w_m]$.

☞ 虽然投影矩阵 P 由子空间 $\mathcal{S}_1$ 和 $\mathcal{S}_2$ 唯一确定, 但其矩阵表示形式 (1.2) 并不唯一.

定理 1.14 (正交投影) 设 $\mathcal{S}_1$ 是内积空间 $\mathcal{S}$ 的一个子空间, $x \in \mathcal{S}$, 则 x 可唯一分解成

$$x = x_1 + x_2, \quad x_1 \in \mathcal{S}_1,\ x_2 \in \mathcal{S}_1^{\perp},$$

其中 x_1 称为 x 在 $\mathcal{S}_1$ 上的**正交投影**.

若 P 是从 $\mathcal{S}$ 沿子空间 $\mathcal{S}_1^{\perp}$ 到子空间 $\mathcal{S}_1$ 上的投影变换, 则称 P 为子空间 $\mathcal{S}_1$ 上的 **正交投影变换** (也称 **正交投影算子** 或 **正交投影矩阵**, 记为 $P_{\mathcal{S}_1}$. 如果 P 不是正交投影变换, 则称其为**斜投影变换** (oblique projector)

定理 1.15 投影矩阵 $P \in \mathbb{R}^{n\times n}$ 是正交投影矩阵的充要条件 $P^T = P$.

设 P 是子空间 $\mathcal{S}_1$ 上的 **正交投影变换**. 令 $v_1, v_2, \ldots, v_m$ 是 $\mathcal{S}_1$ 的一组标准正交基, 则

$$P = VV^T.$$

性质 1.6 设 $P \in \mathbb{R}^{n\times n}$ 是一个正交投影矩阵, 则

$$\|P\|_2 = 1,$$

且对 $\forall x \in \mathbb{R}^n$, 有

$$\|x\|_2^2 = \|Px\|_2^2 + \|(I-P)x\|_2^2.$$

下面是关于正交投影矩阵的一个很重要的应用.

定理 1.16 设 $\mathcal{S}_1$ 是 $\mathbb{R}^n$ 的一个子空间, $z \in \mathbb{R}^n$ 是一个向量. 则最佳逼近问题

$$\min_{x\in\mathcal{S}_1} \|x - z\|_2 \tag{1.3}$$

的唯一解为

$$x^{\star} = P_{\mathcal{S}_1} z.$$

即 $\mathcal{S}_1$ 中距离 z 最近 (2-范数意义下) 的向量是 z 在 $\mathcal{S}_1$ 上的正交投影.

推论 1.17 设矩阵 $A \in \mathbb{R}^{n\times n}$ 对称正定, 向量 $x^{\star} \in \mathcal{S}_1 \subseteq \mathbb{R}^n$. 则 $x^{\star}$ 是最佳逼近问题

$$\min_{x\in\mathcal{S}_1} \|x - z\|_A$$

的解的充要条件是

$$A(x^{\star} - z) \perp \mathcal{S}_1.$$

这里 $\|x - z\|_A \triangleq \|A^{\frac{1}{2}}(x - z)\|_2$.

1.2.6 常见的特殊矩阵

- 实对称矩阵: $A \in \mathbb{R}^{n\times n}$ 且 $A^T = A$;
- Hermite 矩阵 (Hermitian matrix): $A^* = A$;
- 反对称矩阵 (skew-symmetric matrix): $A^T = -A$;
- 反 Hermite 矩阵 (skew-Hermitian matrix): $A^* = -A$;
- 正规矩阵 (normal matrix): $A^*A = AA^*$;
- 酉矩阵 (unitary matrix): $U^*U = UU^* = I$;
- 正交矩阵 (orthogonal matrix): $Q \in \mathbb{R}^{n\times n}$ 且 $Q^TQ = QQ^T = I$;
- 对角矩阵 (diagonal matrix): $a_{ij} = 0$ for $i \neq j$, 记为: $A = \mathrm{diag}(a_{11}, a_{22}, \ldots, a_{nn})$;
- 三对角矩阵 (tridiagonal matrix): $a_{ij} = 0$ for $|i-j| > 1$, 记为: $A = \mathrm{tridiag}(a_{i,i-1}, a_{i,i}, a_{i,i+1})$;
- 上三角矩阵 (upper-triangular matrix): $a_{ij} = 0$ for $i > j$;
- 下三角矩阵 (lower-triangular matrix): $a_{ij} = 0$ for $i < j$;
- 带状矩阵 (banded matrix): $a_{ij} \neq 0$ only if $-b_u \leq i-j \leq b_l$, 其中 b_u 和 b_l 为非负整数, 分别称为下带宽和上带宽, $b_u + b_l + 1$ 称为 A 的带宽 (bandwidth);
- 置换矩阵 (或排列矩阵, permutation matrix): 将单位矩阵 I 的列重新排列所得到的矩阵;
- 上 Hessenberg 矩阵 (upper Hessenberg matrix): $a_{ij} = 0$ for $i-j > 1$;
- 下 Hessenberg 矩阵 (lower Hessenberg matrix): $a_{ij} = 0$ for $i-j < 1$;
- 块对角矩阵 (block-diagonal matrices): $A = \mathrm{Diag}(A_{11}, A_{22}, \ldots, A_{kk})$;
- 块三对角矩阵 (block-triangular matrices): $A = \mathrm{Tridiag}(A_{i,i-1}, A_{i,i}, A_{i,i+1})$;
- Toeplitz 矩阵:

$$T = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+1} \\ t_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix}$$

- 循环矩阵 (circulant matrix):

$$C = \begin{bmatrix} c_0 & c_{n-1} & c_{n-2} & \cdots & c_1 \\ c_1 & c_0 & c_{n-1} & \cdots & c_2 \\ c_2 & c_1 & c_0 & \cdots & c_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n-1} & c_{n-2} & c_{n-3} & \cdots & c_0 \end{bmatrix}$$

- Hankel 矩阵:

$$H = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-2} & h_{n-1} \\ h_1 & \iddots & \iddots & \iddots & h_n \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ h_{n-2} & \iddots & \iddots & \iddots & h_{2n-2} \\ h_{n-1} & h_n & \cdots & h_{2n-2} & h_{2n-1} \end{bmatrix}$$

1.2.7 对角占优矩阵

定义 1.12 设 $A \in \mathbb{R}^{n\times n}$, 若

$$|a_{ii}| \geq \sum_{j\neq i} |a_{ij}|$$

对所有 $i = 1, 2, \ldots, n$ 都成立, 且至少有一个不等式严格成立, 则称 A 为**弱行对角占优**的. 若对所有 $i = 1, 2, \ldots, n$ 不等式都严格成立, 则称 A 是**严格行对角占优**的. 通常简称为**弱对角占优**和**严格对角占优**.

☞ 类似地, 可以定义**弱列对角占优** 和 **严格列对角占优**.

定理 1.18 若 $A \in \mathbb{R}^{n\times n}$ 是严格对角占优矩阵, 则 A 非奇异.

证明. 我们使用反证法. 假设 A 奇异, 即 $Ax = 0$ 存在非零解, 不妨设为 $x = [x_1, x_2, \ldots, x_n]^T$. 不失一般性, 设 $\|x\|_\infty = |x_k|$, 则 $|x_k| > 0$. 考察 $Ax = 0$ 的第 k 个方程:

$$a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n = 0.$$

可得

$$|a_{kk}| = \frac{1}{|x_k|}\left|\sum_{j=1,j\neq i}^{n} a_{kj}x_j\right| \leq \sum_{j=1,j\neq i}^{n} |a_{kj}| \cdot \frac{|x_j|}{|x_k|} \leq \sum_{j=1,j\neq i}^{n} |a_{kj}|,$$

这与 A 严格对角占优矛盾. 所以 A 非奇异. □

1.2.8 特征值与特征向量

定义 1.13 设 $A \in \mathbb{R}^{n\times n}$, 称 $p(\lambda) = \det(A - \lambda I)$ 为 A 的**特征多项式**, 其零点就是 A 的**特征值**.

定义 1.14 设 $A \in \mathbb{R}^{n\times n}$. 若存在 $\lambda \in \mathbb{C}$ 和非零向量 $x, y \in \mathbb{C}^n$, 满足

$$Ax = \lambda x, \qquad y^*A = \lambda y^*,$$

则称 λ 为 A 的**特征值**, x 为 A 对应于 λ 的（右）**特征向量**, y 为 A 对应于 λ 的**左特征向量**, 并称 (λ, x) 为 A 的一个**特征对** *(eigenpair)*.

关于特征值的几个说明:

- 只有当 A 是方阵时, 才具有特征值与特征向量;
- 实矩阵的特征值与特征向量有可能是复的;
- n 阶矩阵总是存在 n 个特征值（其中可能有相等的）;
- 特征值有代数重数和几何重数;
- 相似变换不改变矩阵的特征值;
- 矩阵 A 的所有特征值组成的集合称为 A 的谱, 通常记为 $\sigma(A)$.

定义 1.15 设 $A \in \mathbb{R}^{n\times n}$. 若存在一个非奇异矩阵 $X \in \mathbb{C}^{n\times n}$, 使得

$$X^{-1}AX = \Lambda, \tag{1.4}$$

其中 $\Lambda \in \mathbb{C}^{n\times n}$ 是对角矩阵. 则称 A 是**可对角化**的, 矩阵 Λ 的对角线元素即为 A 的特征值, 分解 (1.4) 称为矩阵 A 的**特征值分解**或**谱分解**.

定理 1.19 设 $A \in \mathbb{R}^{n\times n}$. 则

(1) A 可对角化当且仅当 A 有 n 个线性无关的特征向量;

(2) A 可对角化当且仅当 A 的所有特征值的代数重数与几何重数都相等;

(3) 若 A 有 n 个互不相等的特征值, 则 A 可对角化.

定理 1.20 (Bendixson) 设 $A \in \mathbb{C}^{n\times n}$, 令 $H = \dfrac{1}{2}(A + A^*)$, $S = \dfrac{1}{2}(A - A^*)$. 则有

$$\begin{aligned} \lambda_{min}(H) &\le \mathrm{Re}(\lambda(A)) \le \lambda_{max}(H), \\ \lambda_{min}(iS) &\le \mathrm{Im}(\lambda(A)) \le \lambda_{max}(iS), \end{aligned}$$

其中 $\mathrm{Re}(\cdot)$ 和 $\mathrm{Im}(\cdot)$ 分别表示实部和虚部.

这个定理告诉我们, 一个矩阵的特征值的实部的取值范围由其 Hermite 部分确定, 而虚部则由其反 Hermite 部分确定.

定理 1.21 矩阵的特征值关于矩阵元素是连续的. 即当矩阵的元素发生变化时, 其特征值的变化是连续的.

该结论可以通过多项式零点关于多项式系数的连续性得到.

Gerschgorin 圆盘定理

设 $A = [a_{ij}] \in \mathbb{C}^{n\times n}$, 定义集合

$$\mathcal{D}_i \triangleq \left\{ z \in \mathbb{C} \,:\, |z - a_{ii}| \le \sum_{j=1, j\ne i}^{n} |a_{ij}| \right\}, \quad i = 1, 2, \ldots, n. \tag{1.5}$$

这就是 A 的 n 个 Gerschgorin 圆盘.

定理 1.22 (Gerschgorin 圆盘定理) 设 $A = [a_{ij}] \in \mathbb{C}^{n\times n}$. 则 A 的所有特征值都包含在 A 的 *Gerschgorin* 圆盘的并集中, 即 $\sigma(A) \subset \bigcup_{i=1}^{n} \mathcal{D}_i$.

证明. 设 λ 是 A 的特征值, 对应的非零特征向量为 $x = [x_1, x_2, \ldots, x_n]^T \in \mathbb{C}^n$, 即 $Ax = \lambda x$. 不失一般性, 设 $\|x\|_\infty = |x_i|$, 则 $|x_i| > 0$. 考察 $Ax = \lambda x$ 的第 i 个方程可得

$$\lambda x_i - a_{ii}x_i = \sum_{j=1, j\ne i}^{n} a_{ij}x_j.$$

因此

$$|\lambda - a_{ii}| = \frac{1}{|x_i|} \cdot \left| \sum_{j=1, j\neq i}^{n} a_{ij} x_j \right| \leq \sum_{j=1, j\neq i}^{n} |a_{ij}| \cdot \frac{|x_j|}{|x_i|} \leq \sum_{j=1, j\neq i}^{n} |a_{ij}|.$$

所以 $\lambda \in \mathcal{D}_i$. □

将 A 的非对角线元素换成 τa_{ij}, 其中 $0 \leq \tau \leq 1$, 并利用特征值关于矩阵元素的连续性, 我们就可以得到下面的结论.

定理 1.23 设 $A = [a_{ij}] \in \mathbb{C}^{n\times n}$. 如果 $\bigcup_{i=1}^{n} \mathcal{D}_i$ 可分解成两个不相交的子集 S 和 T, 即

$$\bigcup_{i=1}^{n} \mathcal{D}_i = S \bigcup T \quad \text{且} \quad S \bigcap T = \emptyset.$$

假定 S 由 k 的圆盘组成, 而 T 由其它 $n-k$ 个圆盘组成. 则 S 中恰好包含 A 的 k 个特征值 (重特征值按重数计算), 而 T 中则包含 A 的其它 $n-k$ 个特征值.

定理 1.24 设 $A = [a_{ij}] \in \mathbb{C}^{n\times n}$ 不可约. 如 A 的一个特征值 λ 在 $\bigcup_{i=1}^{n} \mathcal{D}_i$ 的边界上, 则它必定在所有圆盘 $\mathcal{D}_i$ 的边界上 (即圆周上).

证明. 参见 [47]. □

推论 1.25 设 $A = [a_{ij}] \in \mathbb{C}^{n\times n}$ 不可约. 如 λ 在 $\bigcup_{i=1}^{n} \mathcal{D}_i$ 的边界上, 但至少有一个圆盘的圆周不经过 λ, 则它必定不是 A 的特征值.

引理 1.3 若 $A \in \mathbb{R}^{n\times n}$ 是不可约弱对角占优矩阵, 则 A 非奇异.

证明. 使用 Gerschgorin 圆盘定理, 可参见 [47]. □

1.2.9 对称正定矩阵

定义 1.16 设 $A \in \mathbb{C}^{n\times n}$.

- 若对所有非零向量 $x \in \mathbb{C}^n$ 有 $x^*Ax \geq 0$, 则称 A 为 Hermite 半正定; 进一步, 若对所有非零向量 $x \in \mathbb{C}^n$ 有 $x^*Ax > 0$, 则称 A 为 Hermite 正定;
- 若对所有非零向量 $x \in \mathbb{C}^n$ 有 $\mathrm{Re}(x^*Ax) \geq 0$, 则称 A 是半正定的; 进一步, 若对所有非零向量 $x \in \mathbb{C}^n$ 有 $\mathrm{Re}(x^*Ax) > 0$, 则称 A 是正定的.

☞ 若对所有向量 $x \in \mathbb{C}^n$ 有 $x^*Ax \in \mathbb{R}$, 则 $A^* = A$. 因此, 若 A 是 Hermite (半) 正定的, 则 A 必定是 Hermite 矩阵.

☞ 正定和半正定矩阵不一定对称或 Hermite .

定理 1.26 设 $A \in \mathbb{C}^{n\times n}$. 则 A 正定 (半正定) 的充要条件是矩阵 $H = \frac{1}{2}(A + A^*)$ 正定 (半正定).

定理 1.27 设 $A \in \mathbb{R}^{n\times n}$. 则 A 正定 (半正定) 的充要条件是对任意非零向量 $x \in \mathbb{R}^n$ 有 $x^T Ax > 0$ ($x^T Ax \geq 0$).

下面, 我们列出关于 Hermite (半) 正定矩阵的一些常用性质.

定理 1.28 设 $A \in \mathbb{C}^{n\times n}$ 是一个 *Hermite* 半正定矩阵, k 是一个给定的正整数. 则存在一个唯一的 *Hermite* 半正定矩阵 $B \in \mathbb{C}^{n\times n}$ 使得

$$B^k = A.$$

同时, 我们还有下面的性质:

(1) $BA = AB$, 且存在一个多项式 $p(t)$ 使得 $B = p(A)$;

(2) $\mathrm{rank}(B) = \mathrm{rank}(A)$, 因此, 若 A 是正定的, 则 B 也正定;

(3) 如果 A 是实矩阵的, 则 B 也是实矩阵.

Hermite 正定矩阵与内积之间有下面的关系.

定理 1.29 设 $(\cdot,\cdot)$ 是 $\mathbb{C}^n$ 上的一个内积, 则存在一个 *Hermite* 正定矩阵 $A \in \mathbb{C}^{n\times n}$ 使得

$$(x, y) = y^* Ax.$$

反之, 若 $A \in \mathbb{C}^{n\times n}$ 是 *Hermite* 正定矩阵, 则

$$f(x, y) \triangleq y^* Ax$$

是 $\mathbb{C}^n$ 上的一个内积.

☞ 上述性质在 $\mathbb{R}^n$ 中也成立.

1.2.10 不变子空间

定义 1.17 设 $A \in \mathbb{R}^{n\times n}$, 子空间 $\mathcal{S} \subseteq \mathbb{R}^n$. 若 $A\mathcal{S} \subseteq \mathcal{S}$, 即对任意 $x \in \mathcal{S}$, 都有 $Ax \in \mathcal{S}$, 则称 $\mathcal{S}$ 为 A 的一个不变子空间.

定理 1.30 设 $x_1, x_2, \ldots, x_m$ 是 A 的一组线性无关的特征向量, 则 $\mathrm{span}\{x_1, x_2, \ldots, x_m\}$ 是 A 的一个 m 维不变子空间.

定理 1.31 设 $A \in \mathbb{R}^{n\times n}$, $X \in \mathbb{R}^{n\times k}$ 且 $\mathrm{rank}(X) = k$. 则 $\mathrm{span}(X)$ 是 A 的不变子空间的充要条件是存在一个矩阵 $B \in \mathbb{R}^{k\times k}$ 使得

$$AX = XB,$$

此时, B 的特征值都是 A 的特征值.

证明. 我们首先证明必要性. 设 $\text{span}(X)$ 是 A 的不变子空间, 并设 $X = [x_1, x_2, \ldots, x_k]$, 则 $Ax_j \in \text{span}(X)$. 又 $\text{rank}(X) = k$, 故向量组 $\{x_1, x_2, \ldots, x_k\}$ 构成子空间 $\text{span}(X)$ 的一组基. 所以有

$$Ax_j = b_{1j}x_1 + b_{2j}x_2 + \cdots + b_{kj}x_k, \quad j = 1, 2, \ldots, k,$$

其中 $b_{ij} \in \mathbb{R}$ 是线性表出系数. 将上式写成矩阵形式即为

$$AX = XB \qquad \text{其中} \quad B = [b_{ij}] \in \mathbb{R}^{k\times k}.$$

其次证明充分性. 设存在矩阵 $B \in \mathbb{R}^{k\times k}$, 使得 $AX = XB$. 则 Ax_j 为 $x_1, x_2, \ldots, x_k$ 的线性组合. 又 $\{x_1, x_2, \ldots, x_k\}$ 为 $\text{span}(X)$ 的一组基, 所以对任意 $x \in \text{span}(X)$ 都有 $Ax \in \text{span}(X)$, 即 $\text{span}(X)$ 是 A 的一个不变子空间.

下面证明 B 的特征值都是 A 的特征值. 将 X 扩充成一个非奇异的方阵, 即存在矩阵 $\tilde{X} \in \mathbb{R}^{n\times(n-k)}$, 使得 $Y = [X, \tilde{X}] \in \mathbb{R}^{n\times n}$ 非奇异. 将 Y^{-1} 写成分块形式: $Y^{-1} = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}$, 其中 $Z_1 \in \mathbb{R}^{k\times n}$, $Z_2 \in \mathbb{R}^{(n-k)\times n}$. 由等式 $Y^{-1}Y = I_{n\times n}$ 可得 $Z_1X = I_{k\times k}$, $Z_2X = 0$. 又 $AX = XB$, 所以

$$Y^{-1}AY = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} A[X, \tilde{X}] = \begin{bmatrix} Z_1AX & Z_1A\tilde{X} \\ Z_2AX & Z_2A\tilde{X} \end{bmatrix} = \begin{bmatrix} Z_1XB & Z_1A\tilde{X} \\ Z_2XB & Z_2A\tilde{X} \end{bmatrix} = \begin{bmatrix} B & Z_1A\tilde{X} \\ 0 & Z_2A\tilde{X} \end{bmatrix}.$$

因此 B 的特征值都是 $Y^{-1}AY$ 的特征值. 由于 A 与 $Y^{-1}AY$ 相似, 它们具有相同的特征值. 定理结论成立. □

推论 1.32 设 $A \in \mathbb{R}^{n\times n}$, $X \in \mathbb{R}^{n\times k}$ 且 $\text{rank}(X) = k$. 若存在一个矩阵 $B \in \mathbb{R}^{k\times k}$ 使得 $AX = XB$, 则 (λ, v) 是 B 的一个特征对当且仅当 (λ, Xv) 是 A 的一个特征对.

1.2.11 Jordan 标准型

在计算矩阵的特征值时, 一个基本的思想是通过相似变换, 将其转化成一个形式尽可能简单的矩阵, 使得其特征值更易于计算. 在这里, 我们介绍两个非常有用的特殊矩阵: **Jordan 标准型**和 **Schur 标准型**.

定理 1.33 设 $A \in \mathbb{R}^{n\times n}$, 则存在非奇异矩阵 $X \in \mathbb{C}^{n\times n}$, 使得

$$X^{-1}AX = \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_p \end{bmatrix},$$

其中 J_i 的维数等于 λ_i 的代数重数, 且具有下面的结构

$$J_i = \begin{bmatrix} J_{i1} & & & \\ & J_{i2} & & \\ & & \ddots & \\ & & & J_{i\nu_i} \end{bmatrix}, \quad J_{ik} = \begin{bmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \lambda_i & 1 \\ & & & \lambda_i \end{bmatrix},$$

这里的 ν_i 为 λ_i 的几何重数, J_{ik} 称为 Jordan 块, 每个 *Jordan* 块对应于一个特征向量.

Jordan 标准型具有以下性质:

- Jordan 块的个数等于 A 的线性无关的特征向量的个数;
- A 可对角化的充要条件是每个 Jordan 块都是 1×1 的, 此时 X 的列向量就是 A 的特征向量;
- 所有可对角化矩阵组成的集合在所有矩阵组成的集合中是稠密的;
- A 是正规矩阵 ($A^TA = AA^T$) 的充要条件是 A 可对角化且 X 是酉矩阵.

Jordan 标准型在理论研究中非常有用, 但数值计算比较困难, 目前还没有找到十分稳定的数值算法. 下面我们介绍一个比较实用的标准型: Schur 标准型.

1.2.12 Schur 标准型

定理 1.34 设 $A \in \mathbb{C}^{n\times n}$, 则存在一个酉矩阵 $U \in \mathbb{C}^{n\times n}$ 使得

$$U^*AU = \begin{bmatrix} \lambda_1 & r_{12} & \cdots & r_{1n} \\ 0 & \lambda_2 & \cdots & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \triangleq R \quad 或 \quad A = URU^*, \tag{1.6}$$

其中 $\lambda_1, \lambda_2, \ldots, \lambda_n$ 是 A 的特征值 (可以按任意顺序排列).

证明. 我们对 n 使用归纳法.

当 $n = 1$ 时, 结论显然成立.

假设结论对所有阶数不超过 $n-1$ 的矩阵都成立. 考虑 n 阶矩阵 $A \in \mathbb{C}^{n\times n}$. 设 λ 是 A 的一个特征值, 其对应的特征向量为 $x \in \mathbb{C}^n$, 且 $\|x\|_2 = 1$. 构造一个以 x 为第一列的酉矩阵 $X = [x, \tilde{X}]$. 于是

$$X^*AX = \begin{bmatrix} x^* \\ \tilde{X}^* \end{bmatrix} A\,[x, \tilde{X}] = \begin{bmatrix} x^*Ax & x^*A\tilde{X} \\ \tilde{X}^*Ax & \tilde{X}^*A\tilde{X} \end{bmatrix}.$$

因为 $x^*Ax = \lambda x^*x = \lambda$, 且 $\tilde{X}^*Ax = \tilde{X}^*(\lambda x) = \lambda\tilde{X}^*x = 0$, 故

$$X^*AX = \begin{bmatrix} \lambda & x^*A\tilde{X} \\ 0 & \tilde{X}^*A\tilde{X} \end{bmatrix} \triangleq \begin{bmatrix} \lambda & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix},$$

其中 $\tilde{A}_{22} \in \mathbb{C}^{(n-1)\times(n-1)}$. 根据归纳假设, 存在酉矩阵 $\tilde{U} \in \mathbb{C}^{(n-1)\times(n-1)}$, 使得 $\tilde{U}^*A_{22}\tilde{U} = \tilde{R} \in \mathbb{C}^{(n-1)\times(n-1)}$ 是一个上三角矩阵. 令

$$U = X \begin{bmatrix} 1 & 0 \\ 0 & \tilde{U} \end{bmatrix},$$

则有

$$
\begin{aligned}
U^*AU &= \begin{bmatrix} 1 & 0 \\ 0 & \tilde{U}^* \end{bmatrix} X^*AX \begin{bmatrix} 1 & 0 \\ 0 & \tilde{U} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 \\ 0 & \tilde{U}^* \end{bmatrix} \begin{bmatrix} \lambda & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{U} \end{bmatrix} \\
&= \begin{bmatrix} \lambda & \tilde{A}_{12}\tilde{U} \\ 0 & \tilde{U}^*\tilde{A}_{22}\tilde{U} \end{bmatrix} \\
&= \begin{bmatrix} \lambda & \tilde{A}_{12}\tilde{U} \\ 0 & \tilde{R} \end{bmatrix} \triangleq R.
\end{aligned}
$$

由于 $\tilde{R}$ 是上三角矩阵, 故 R 也是一个上三角矩阵, 其对角线元素即为 A 的特征值.

由归纳法可知, 定理结论成立. □

关于 Schur 标准型的几点说明:

- Schur 标准型可以说是酉相似变化下的最简形式;
- 定理中的 U 和 R 不是唯一的, R 的对角线元素可以按任意顺序排列.

推论 1.35 设 $A \in \mathbb{C}^{n\times n}$, 则

- A 是正规矩阵当且仅当 (1.6) 中的 R 是对角矩阵;
- A 是 Hermite 矩阵当且仅当 (1.6) 中的 R 是实对角矩阵.

众所周知, 当 A 是实矩阵时, 其特征值和特征向量仍可能是复的. 但在实际计算中, 我们希望避免复数运算. 下面我们给出**实 Schur 标准型** (或**拟 Schur 标准型**).

定理 1.36 设 $A \in \mathbb{R}^{n\times n}$, 则存在正交矩阵 $Q \in \mathbb{R}^{n\times n}$, 使得

$$
Q^T AQ = T,
$$

其中 $T \in \mathbb{R}^{n\times n}$ 是**拟上三角矩阵**, 即 T 是块上三角的, 且对角块为 1×1 或 2×2 的块矩阵. 若对角块是 1×1 的, 则其就是 A 的一个特征值, 若对角块是 2×2 的, 则其特征值是 A 的一对共轭复特征值.

证明. 同样可以使用归纳法.

设 λ 是 A 的一个特征值. 若 λ 是实的, 则存在一个对应的实特征向量, 后面的证明与定理 1.34 的证明类似.

若 λ 是复数, 设其对应的单位复特征向量为 u. 由于

$$
\bar{\lambda}\bar{u} = \overline{\lambda u} = \overline{Au} = \bar{A}\bar{u} = A\bar{u},
$$

故 $(\bar{\lambda}, \bar{u})$ 也是 A 的一个特征对. 令

$$
\tilde{u} = \frac{1}{2}(u + \bar{u}), \qquad \tilde{v} = \frac{1}{2i}(u - \bar{u}),
$$

即 $\tilde{u}, \tilde{v}$ 分别为 u 的实部与虚部, 故 $\tilde{u} \in \mathbb{R}^n, \tilde{v} \in \mathbb{R}^n$. 所以由定理 1.30 可知, $\mathrm{span}\{\tilde{u}, \tilde{v}\} = \mathrm{span}\{u, \bar{u}\}$ 是 A 的一个不变子空间. 将 $\{\tilde{u}, \tilde{v}\}$ 进行正交化, 即存在列正交矩阵 $\tilde{Q} \in \mathbb{R}^{n\times 2}$ 和非奇异上三角矩阵 $R \in \mathbb{R}^{2\times 2}$, 使得 $[\tilde{u}, \tilde{v}] = \tilde{Q}\tilde{R}$. 则 $\mathrm{span}\{\tilde{Q}\} = \mathrm{span}\{\tilde{u}, \tilde{v}\}$ 也是 A 的不变子空间, 由定理 1.31 可知, 存在矩阵 $B \in \mathbb{R}^{2\times 2}$ 使得 $A\tilde{Q} = \tilde{Q}B$. 将 $\tilde{Q}$ 扩充成一个正交矩阵, 即存在矩阵 $\hat{Q} \in \mathbb{R}^{n\times(n-2)}$ 使得 $\left[\tilde{Q}, \hat{Q}\right]$ 是正交矩阵. 于是有

$$\left[\tilde{Q}, \hat{Q}\right]^T A \left[\tilde{Q}, \hat{Q}\right] = \begin{bmatrix} \tilde{Q}^T A\tilde{Q} & \tilde{Q}^T A\hat{Q} \\ \hat{Q}^T A\tilde{Q} & \hat{Q}^T A\hat{Q} \end{bmatrix} = \begin{bmatrix} B & \tilde{Q}^T A\hat{Q} \\ 0 & \hat{Q}^T A\hat{Q} \end{bmatrix},$$

其中 $\hat{Q}^T A\hat{Q} \in \mathbb{R}^{(n-2)\times(n-2)}$. 对 $\hat{Q}^T A\hat{Q}$ 使用归纳假设, 即可证明定理结论成立. □

若 A 的特征值都是实数, 则可得

推论 1.37 设 $A \in \mathbb{R}^{n\times n}$ 的特征值都是实的, 则存在正交矩阵 $Q \in \mathbb{R}^{n\times n}$ 和上三角矩阵 $R \in \mathbb{R}^{n\times n}$ 使得

$$Q^T AQ = R,$$

其中 R 的对角线元素即为 A 的特征值.

1.2.13 Kronecker 积

定义 1.18 设 $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{p\times q}$, 则 A 与 B 的 Kronecker 积定义为

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix} \in \mathbb{C}^{mp\times nq}.$$

☞ Kronecker 积也称为**直积**, 或**张量积**.

☞ 任意两个矩阵都存在 Kronecker 积, 且 $A \otimes B$ 和 $B \otimes A$ 是同阶矩阵, 但通常 $A \otimes B \neq B \otimes A$.

定理 1.38 矩阵的 *Kronecker* 积有以下性质:

(1) $(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha(A \otimes B), \quad \forall\, \alpha \in \mathbb{C}$;

(2) $(A \otimes B)^T = A^T \otimes B^T, \quad (A \otimes B)^* = A^* \otimes B^*$;

(3) $(A \otimes B) \otimes C = A \otimes (B \otimes C)$;

(4) $(A + B) \otimes C = A \otimes C + B \otimes C$;

(5) $A \otimes (B + C) = A \otimes B + A \otimes C$;

(6) 混合积: $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$

(7) $(A_1 \otimes A_2 \otimes \cdots \otimes A_k)(B_1 \otimes B_2 \otimes \cdots \otimes B_k) = (A_1B_1) \otimes (A_2B_2) \otimes \cdots \otimes (A_kB_k)$;

(8) $(A_1 \otimes B_1)(A_2 \otimes B_2) \cdots (A_k \otimes B_k) = (A_1A_2 \cdots A_k) \otimes (B_1B_2 \cdots B_k)$;

(9) $\mathrm{rank}(A \otimes B) = \mathrm{rank}(A)\,\mathrm{rank}(B)$;

定理 1.39 设 $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, 并设 (λ, x) 和 (μ, y) 分别是 A 和 B 的一个特征对, 则 $(\lambda\mu, x\otimes y)$ 是 $A\otimes B$ 的一个特征对. 由此可知, $B\otimes A$ 与 $A\otimes B$ 具有相同的特征值.

定理 1.40 设 $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, 则

(1) $\mathrm{tr}(A\otimes B) = \mathrm{tr}(A)\mathrm{tr}(B)$;

(2) $\det(A\otimes B) = \det(A)^n \det(B)^m$;

(3) $A\otimes I_n + I_m \otimes B$ 的特征值为 $\lambda_i + \mu_j$, 其中 λ_i 和 μ_j 分别为 A 和 B 的特征值;

(4) 若 A 和 B 都非奇异, 则 $(A\otimes B)^{-1} = A^{-1}\otimes B^{-1}$;

推论 1.41 设 $A = Q_1\Lambda_1 Q_1^{-1}$, $B = Q_2\Lambda_2 Q_2^{-1}$, 则

$$A\otimes B = (Q_1\otimes Q_2)(\Lambda_1\otimes\Lambda_2)(Q_1\otimes Q_2)^{-1}.$$

定理 1.42 设 $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, 则存在 $m+n$ 阶置换矩阵 P 使得

$$P^T(A\otimes B)P = B\otimes A.$$

定理 1.43 设矩阵 $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m\times n}$, 记 $\mathrm{vec}(X)$ 为 X 按列拉成的 mn 维列向量, 即

$$\mathrm{vec}(X) = [x_1^T, x_2^T, \ldots, x_N^T]^T,$$

则有

$$\mathrm{vec}(AX) = (I\otimes A)\mathrm{vec}(X), \quad \mathrm{vec}(XB) = (B^T\otimes I)\mathrm{vec}(X),$$

以及

$$(A\otimes B)\mathrm{vec}(X) = \mathrm{vec}(BXA^T).$$

定理 1.44 矩阵方程

$$AX + XB = D$$

等价于代数方程

$$(I\otimes A + B^T\otimes I)\mathrm{vec}(X) = \mathrm{vec}(D).$$

1.3 数值计算中的误差

数值方法的特点之一就是所求得的解是近似解, 总是存在一定的误差. 因此, 误差分析是数值分析中一个很重要的课题.

误差是人们用来描述数值计算中近似解的精确程度, 是科学计算中的一个十分重要的概念.

误差大致可分为以下几种类型:

- 模型误差: 数学模型是对实际问题的数学描述, 它往往是抓住问题的主要因素而略去次要因素, 因此, 它是实际问题的一个近似.
- 观测误差: 在数学模型中通常包含一些参量 (数据), 这些参量的值一般都是通过测量或实验的方法所得到的, 因此也存在误差.
- 截断误差: 也称方法误差, 在对数学模型进行数值求解时, 需要做一些近似, 如对导数离散时可用差商代替.
- 舍入误差: 由于机器字长有限, 由于机器字长有限, 计算机对浮点数的表示和算术运算都存在一定的误差.

在数值分析中, 我们总是假定数学模型和所给的数据都是准确的, 因而不考虑模型误差和观测误差, 主要研究截断误差和舍入误差对计算结果的影响.

例 1.6 近似计算 $\int_0^1 e^{-x^2}\,\mathrm{d}x$ 的值.

解. 这里我们采用 Taylor 展开, 即

$$\begin{aligned}\int_0^1 e^{-x^2}\,\mathrm{d}x &= \int_0^1 \left(x - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \frac{x^8}{4!} - \cdots\right) \\ &= 1 - \frac{1}{3} + \frac{1}{2!}\times\frac{1}{5} - \frac{1}{3!}\times\frac{1}{7} + \frac{1}{4!}\times\frac{1}{9} - \cdots \\ &\triangleq S_4 + R_4\end{aligned}$$

其中 S_4 为前四项的部分和, R_4 为剩余部分. 如果我们以 S_4 作为定积分的近似值, 则 R_4 就是由此产生的误差, 这种误差就称为截断误差.

在计算 S_4 的值, 假定我们保留小数点后 4 位有效数字, 则

$$S_4 = 1 - \frac{1}{3} + \frac{1}{10} - \frac{1}{42} \approx 1 - 0.3333 + 0.1000 - 0.0238 = 0.7429$$

这就是我们最后得到的近似值. 这里, 在计算 S_4 时所产生的误差就是舍入误差. □

1.3.1 误差与有效数字

定义 1.19 设 x^* 是精确值, x 是其近似值, 则绝对误差 e 和相对误差 e_r 分别定义为

$$e = x - x^* \qquad \text{和} \qquad e_r = \frac{x - x^*}{x^*}.$$

若存在 $\varepsilon > 0$ 满足 $|e| = |x - x^*| \le \varepsilon$, 则称 ε 为绝对误差限, 简称误差限. 类似地, 若存在 $\varepsilon_r > 0$ 满足 $|e_r| \le \varepsilon_r$, 则称 ε_r 为相对误差限.

几点说明:

- 绝对误差可能为正, 也可能为负;
- 绝对误差越小越具有参考价, 但绝对误差却不能很好地表示近似值的精确程度;
- 近似值的精确程度取决于相对误差的大小;
- 实际计算中我们所能估计的通常是误差限或相对误差限;
- 由于真值难以求出, 通常也使用下面的定义作为相对误差限

$$e_r = \frac{x - x^*}{x}.$$

- 工程中通常用下面的表达式来刻画近似值的精度:

$$x^* = x \pm \varepsilon,$$

表示精确值在区间 $[x - \varepsilon, x + \varepsilon]$ 中.

定义 1.20 若近似值 x 的误差限是某一位的半个单位, 且该位到 x 的第一位非零数字共有 n 位, 则称 x 有 n 位 **有效数字**.

关于有效数字的判断, 我们可以使用下面的方法.

性质 1.7 设 x 是 x^* 的近似值, 若 x 可表示为

$$x = \pm 0.a_1 a_2 \ldots a_n \ldots \times 10^m,$$

其中 a_i 是 0 到 9 中的数字, 且 $a_1 \neq 0$. 若

$$|x - x^*| \leq 0.5 \times 10^{m-n},$$

则 x 至少有 n 位有效数字.

换而言之, 若 $|x - x^*| \leq 0.5 \times 10^k$, 则 x 至少有 $m - k$ 位有效数字.

例 1.7 设 $x_1 = 3.1415$ 和 $x_2 = 3.1416$ 是 $\pi = 3.14159265...$ 的近似值, 则 x_1 有 4 位有效数字, 而 x_2 有 5 位有效数字.

例 1.8 根据四舍五入原则, 写出下列各数的具有 5 位有效数字的近似值:

$$187.9325, \quad 0.03785551, \quad 8.000033.$$

解. 这三个数的具有 5 位有效数字的近似值分别为: 187.93, 0.037856, 8.0000. □

有两点需要注意的是:

- 按四舍五入原则得到的数字是有效数字;
- 一个数末尾的 0 不可以随意添加或省略.

定理 1.45 (有效数字与相对误差限) 设 x 是 x^* 的近似值, 若 x 可表示为

$$x = \pm 0.a_1 a_2 \ldots a_n \ldots \times 10^m,$$

其中 a_i 是 0 到 9 中的数字, 且 $a_1 \neq 0$. 若 x 具有 n 位有效数字, 则其相对误差限满足

$$\varepsilon_r \leq \frac{1}{2a_1} \times 10^{-n+1}.$$

反之, 若 x 的相对误差限满足

$$\varepsilon_r \leq \frac{1}{2(a_1+1)} \times 10^{-n+1}.$$

则 x 至少有 n 位有效数字.

证明. 由 x 的表达式可知

$$a_1 \times 10^{m-1} \leq |x| \leq (a_1+1) \times 10^{m-1}.$$

若 x 具有 n 位有效数字, 则

$$\varepsilon_r = \frac{|x-x^*|}{|x|} \leq \frac{0.5 \times 10^{m-n}}{a_1 \times 10^{m-1}} = \frac{1}{2a_1} \times 10^{-n+1}.$$

反之, 若 $\varepsilon_r \leq \frac{1}{2(a_1+1)} \times 10^{-n+1}$, 则

$$|x-x^*| = |x| \cdot \varepsilon_r \leq 0.5 \times 10^{m-n}.$$

故 x 至少有 n 位有效数字. □

从这个定理可以看出, 有效数字与相对误差是紧密相关的: 有效数字越多, 相对误差就越小; 反之, 相对误差越小, 有效数字就越多.

基本算术运算的误差估计

误差估计主要是指如何估计误差限或相对误差限. 我们用 $\varepsilon(x)$ 表示 x 的误差限, 则有

$$\begin{aligned}
\varepsilon(x_1 \pm x_2) &\leq \varepsilon(x_1) + \varepsilon(x_2), \\
\varepsilon(x_1 x_2) &\leq |x_2|\varepsilon(x_1) + |x_1|\varepsilon(x_2), \\
\varepsilon\left(\frac{x_1}{x_2}\right) &\leq \frac{|x_1|\varepsilon(x_1) + |x_1|\varepsilon(x_2)}{|x_2|^2}.
\end{aligned}$$

函数求值的误差估计

一般地, 设 $f(x)$ 是可微函数, x 为 x^* 的近似值, 则由 Taylor 公式可知, 存在 ξ 使得

$$f(x) - f(x^*) = f'(x^*)(x-x^*) + \frac{1}{2}f''(\xi)(x-x^*)^2.$$

所以有

$$\varepsilon(f(x)) \leq |f'(x^*)|\varepsilon(x) + \frac{|f''(\xi)|}{2}\varepsilon^2(x).$$

当 $|f''(\xi)|$ 与 $|f'(x^*)|$ 的比值不是很大时, 我们可以舍去二次项, 从而得到

$$\varepsilon(f(x)) \lessapprox |f'(x^*)|\varepsilon(x).$$

由于 x^* 通常是不知道的, 所以我们也用 $f'(x)$ 来近似 $f'(x^*)$, 即

$$\varepsilon(f(x)) \lessapprox |f'(x)|\varepsilon(x).$$

关于相对误差限, 我们有如下的估计:

$$\varepsilon_r(f(x)) = \left|\frac{f(x)-f(x^*)}{f(x^*)}\right| \approx \left|\frac{f'(x^*)(x-x^*)}{f(x^*)}\right| = \left|\frac{x^*f'(x^*)}{f(x^*)}\right| \cdot \left|\frac{x-x^*}{x^*}\right| = C_p\varepsilon_r(x),$$

其中 $C_p = \left|\frac{x^*f'(x^*)}{f(x^*)}\right|$ 称为 $f(x)$ 的 **条件数**.

对于多元可微函数 $f(x_1, x_2, \ldots, x_n)$, 设 $x = (x_1, x_2, \ldots, x_n)$ 是 $x = (x_1^*, x_2^*, \ldots, x_n^*)$ 的近似值, 则有

$$\varepsilon(f(x)) \approx \sum_{k=1}^{n} \left|\frac{\partial f(x)}{\partial x_k^*}\right| \varepsilon(x_k).$$

1.3.2 误差分析

- 数值计算中的误差分析很重要, 但也很复杂;
- 在计算过程中, 误差会传播、积累、对消;
- 实际计算中的运算次数通常都在千万次以上, 因此对每一步运算都做误差分析比较不切实际.

误差分析一般可分为定量分析和定性分析.

定量分析

- 主要方法有: 向前误差分析法, 向后误差分析法, 区间误差分析法, 概率分析法等.
- **向前误差分析**: 用输入数据的误差和数值方法本身的误差来分析计算结果的误差.
- **向后误差分析**: 用某个算法计算 $f(x)$, 得到的近似解为 $\tilde{f}$, 假定 $\tilde{f}$ 是 $f(x)$ 对应于某个数据 $\tilde{x}$ 的精确解, 即 $\tilde{f} = f(\tilde{x})$, 分析 $\tilde{x} - x$ 的大小就是向后误差分析.

向后误差分析法 (backward error analysis) 由著名数值分析专家 J. H. Wilkinson 于 1960 年提出, 这是误差理论中最基本的误差分析方法之一.

向后误差分析是一种**先验误差估计**方法, 不仅可以用来讨论算法的稳定性, 还可以用于讨论算法的收敛性.

后验误差估计则是利用得到的数值结果来估计近似解的误差, 如在解方程组时, 可以利用残量来估计解的误差.

图 1.1 Wilkinson (1919-1986)

定性分析

- 目前在数值计算中更关注的是误差的定性分析;
- 定性分析包括研究数学问题的适定性, 数学问题与原问题的相容性, 数值算法的稳定性, 避免扩大误差的准则等;
- 定性分析的核心是原始数据的误差和计算过程中产生的误差对最终计算结果的影响.

☞ 算法有 “优劣” 之分, 问题有 “好坏” 之别, 即使不能定量地估计出最终误差, 但是若能确保计算过程中误差不会被任意放大, 那就能放心地实施计算, 这就是研究定性分析的初衷.

1.3.3 数值稳定性

数值计算中的稳定性包括数学问题的稳定性和数值算法的稳定性.

数学问题的稳定性

如果数学问题满足

(1) 对任意满足一定条件的输入数据, 存在一个解,

(2) 对任意满足一定条件的输入数据, 解是唯一的,

(3) 问题的解关于输入数据是连续的,

则称该数学问题是**适定的** (well-posed), 否则就称为**不适定的** (ill-posed).

> ☞ 如果输入数据的微小扰动会引起输出数据 (即计算结果) 的很大变化 (误差), 则称该数值问题是**病态**的.

例 1.9 解线性方程组 $\begin{cases} x+\alpha y=1 \\ \alpha x+y=0 \end{cases}$

解. 易知当 $\alpha=1$ 时, 方程组无解. 当 $\alpha\neq 1$ 时, 解为

$$x=\frac{1}{1-\alpha^2}, \quad y=\frac{-\alpha}{1-\alpha^2}.$$

当 $\alpha\approx 1$ 时, 解的误差可能会很大. 比如当 $\alpha=0.999$ 时, $x\approx 500.25$. 假定输入数据 α 带有 0.0001 的误差, 即输入数据为 $\alpha^*=0.9991$, 则此时有 $x^*\approx 555.81$, 解的误差约为 55.56, 是输入数据误差的五十多万倍, 因此该问题的病态的. □

病态问题与条件数

设 $f(x)$ 可导, 则其**条件数**定义为

$$C_p=\left|\frac{xf'(x)}{f(x)}\right|.$$

- 一般情况下, 条件数大于 10 时, 就认为问题是病态的;
- 条件数越大问题病态就越严重;
- 病态是问题本身固有的性质, 与数值算法无关;
- 对于病态问题, 选择数值算法时需要谨慎.

算法的稳定性

在计算过程中, 如果误差不增长, 则称该算法是稳定的, 否则为不稳定的.

> ☞ 在数值计算中, 不要采用不稳定的算法!

例 1.10 近似计算

$$S_n=\int_0^1 \frac{x^n}{x+5}\,\mathrm{d}x, \quad n=1,2,\dots,8.$$

解. 通过观察可知

$$S_n+5S_{n-1}=\int_0^1 \frac{x^n+5x^{n-1}}{x+5}\,\mathrm{d}x=\int_0^1 x^{n-1}=\frac{1}{n},$$

因此,

$$S_n=\frac{1}{n}-5S_{n-1}. \tag{1.7}$$

易知 $S_0 = \ln 6 - \ln 5 \approx 0.182$ (保留三位有效数字), 利用上面的递推公式可得 (保留三位有效数字)

$$
\begin{aligned}
&S_1 = 0.0900, \quad S_2 = 0.0500, \quad S_3 = 0.0833, \quad S_4 = -0.166, \\
&S_5 = 1.03, \quad S_6 = -4.98, \quad S_7 = 25.0, \quad S_8 = -125.
\end{aligned}
$$

另一方面, 我们有

$$
\frac{1}{6(n+1)} = \int_0^1 \frac{x^n}{6}\,\mathrm{d}x \leq \int_0^1 \frac{x^n}{x+5}\,\mathrm{d}x \leq \int_0^1 \frac{x^n}{5}\,\mathrm{d}x = \frac{1}{5(n+1)}. \tag{1.8}
$$

因此, 上面计算的 $S_4, \ldots, S_8$ 显然是不对的. 原因是什么呢? 误差!

设 S_n^* 是 S_n 的近似值, 则

$$
e(S_n^*) = S_n^* - S_n = \left(\frac{1}{n} - 5S_{n-1}^*\right) - \left(\frac{1}{n} - 5S_{n-1}\right) = -5(S_{n-1}^* - S_{n-1}) = -5e(S_{n-1}^*).
$$

即误差是以 5 倍速度增长, 这说明计算过程是不稳定的, 因此我们不能使用该算法.

事实上, 递推公式 (1.7) 可以改写为

$$
S_{n-1} = \frac{1}{5n} - \frac{1}{5}S_n.
$$

因此, 我们可以先估计 S_8 的值, 然后通过反向递推, 得到其它值.

我们可以根据 (1.8) 对 S_8 做简单的估计, 即

$$
S_8 \approx \frac{1}{2}\left(\int_0^1 \frac{x^n}{6}\,\mathrm{d}x + \int_0^1 \frac{x^n}{5}\,\mathrm{d}x\right) \approx 0.0204.
$$

于是

$$
\begin{aligned}
&S_7 = 0.0209, \quad S_6 = 0.0244, \quad S_5 = 0.0285, \quad S_4 = 0.0343, \\
&S_3 = 0.0431, \quad S_2 = 0.0580, \quad S_1 = 0.0884, \quad S_0 = 0.182.
\end{aligned}
$$

对比精确值 S_n^* 可知, 此时计算出来的 S_n 精度要好很多.

通过误差分析可知, 误差是以 $\frac{1}{5}$ 的速度减小, 因此计算过程是稳定的. □

☞ 在数值计算中, 误差不可避免, 算法的稳定性是一个非常重要的性质.

算法的稳定性: 通俗地讲, 对于某个给定的算法, 如果输入数据的误差在运算过程不断增长而得不到控制, 那么我们就说该算法是数值不稳定的, 否则就是数值稳定的.

假设输入数据的误差为 e_0, 经 n 次运算后的计算结果的误差为 e_n. 如果 $e_n \approx c_1 n e_0$, 其中 c_1 是与 n 无关的常数, 则称误差是线性增长的. 如果 $e_n \approx c_2 k^n e_0$, 其中 c_2, k 是与 n 无关的常数且 $k > 1$, 则称误差是指数增长的. 如果算法的误差增长是线性的, 则该算法是数值稳定的, 如果算法的误差是指数增长的, 则该算法是数值不稳定的.

显然误差的线性增长是不可避免的, 而指数增长是必须避免的. 在数值计算中, 算法的稳定性是一个非常重要的性质.

定义 1.21 (算法的向后稳定性) 用某个算法来计算 $f(x)$, 得到的近似值为 $\tilde{f}(x)$, 若对任意的 x, 都存在一个"小"的 δx, 使得 $f(x+\delta x)=\tilde{f}(x)$, 则称该算法是**向后稳定的**, 其中 δx 称为 **向后误差**. 这种误差分析方法就是**向后误差分析**.

若一个算法是向后稳定的, 则有

$$|\tilde{f}(x)-f(x)|=|f(x+\delta x)-f(x)|\approx|f'(x)|\cdot|\delta x|.$$

由于 δx 很小, 所以当 $|f'(x)|$ 不是很大时, 误差总是很小. 因此向后稳定是一个好的算法的基本性质.

☞ 向后误差分析将舍入误差归入到截断误差中, 使得误差分析相对简单化.

数值计算注意事项

在用计算机进行数值计算时, 舍入误差不可避免, 但我们要尽可能地减小舍入误差对计算结果的影响. 在计算过程中, 我们应注意以下几点.

(1) 避免相近的数相减

如果两个相近的数相减, 则会损失有效数字, 如 $0.12346-0.12345=0.00001$, 操作数有 5 位有效数字, 但结果却只有 1 为有效数字. 下面给出几个避免相近的数相减的方法:

$$\begin{aligned}
&\sqrt{x+\varepsilon}-\sqrt{\varepsilon}=\frac{\varepsilon}{\sqrt{x+\varepsilon}+\sqrt{x}}\\
&\ln(x+\varepsilon)-\ln(x)=\ln\left(1+\frac{\varepsilon}{x}\right)\\
&1-\cos(x)=2\sin^2\frac{x}{2},\quad |x|\ll 1\\
&\mathrm{e}^x-1=x\left(1+\frac{1}{2}x+\frac{1}{6}x^2+\cdots\right),\quad |x|\ll 1
\end{aligned}$$

(2) 避免数量级相差很大的数相除

可能会产生溢出, 即超出计算机所能表示的数的范围.

(3) 避免大数吃小数

如直接计算 $(10^9-10^{-9}+10^9)/10^{-9}$ 时, 结果可能为 0.

另外, 在对一组数求和时, 应按绝对值从小到大求和.

(4) 简化计算

尽量减少运算次数, 避免误差积累.

(5) 选用稳定的算法

尽可能避免由于算法本身导致的误差增大.

1.4 IEEE 浮点运算标准

数通常以浮点格式表示和参与运算的 (整数除外), 浮点格式是一种数据结构, 用于指定包含浮点数的字段、这些字段的布局及其算术解释. 浮点存储格式指定如何将浮点格式存储在内存中. 自计算机发明以来, 曾出现许多中不同的浮点数表示方式, 但目前最通用的是 IEEE 二进制浮点数算术标准 (IEEE Standard for Binary Floating-Point Arithmetic, 简称 IEEE 754 标准).

IEEE 754 标准的主要起草者是加州大学伯克利分校数学系的 William Kahan 教授, 他帮助 Intel 公司设计了 8087 浮点处理器, 并以此为基础形成了 IEEE 754 标准, Kahan 教授也因此获得了 1987 年的图灵奖.

1.4.1 IEEE 中的浮点数

通常一个浮点数由符号、尾数、基和指数组成, 如:

$$-0.31415926_{10} \times 10^2, \quad 0.10101_2 \times 2^3.$$

这里要求小数点前面为零, 小数点后面的数称为**尾数**. 若尾数的首位数字不为 0 时, 我们称其为**正规数** (或**规范化数**), 否则称为**次正规数** (或**非规范化数**). 如 $0.314_{10} \times 10^2$ 是正规数, 而 $0.00314_{10} \times 10^4$ 是次正规数. 正规化表示方法可以使得每个浮点数的表示方式唯一, 而且可以空出一个位置, 使得表示精度更高.

- IEEE 754 标准中定义了表示浮点数的四种格式:
  - 两种基本的浮点数: **单精度** (32 位字长) 和**双精度** (64 位字长).
    其中单精度格式具有 24 位有效数字 (二进制), 而双精度格式具有 53 位有效数字 (二进制), 相对于十进制来说, 分别是 7 位 ($2^{24} \approx 10^7$) 和 16 位 ($2^{53} \approx 10^{16}$) 有效数字.
  - 两种扩展的浮点数: **单精度扩展**和**双精度扩展**.
    IEEE 754 标准中并未规定扩展格式的精度和大小, 但它指定了最小精度和字长: 单精度扩展需 43 位字长以上, 双精确度扩展需 79 位字长以上 (64 位有效数字). 单精度扩展很少使用, 而对于双精确度扩展, 不同的机器架构中有着不同的规定, 有的为 80 位字长 (如 X86), 有的为 128 位字长 (如 SPARC).
- 一般来说, 描述一个浮点数的三个基本要素为:
  - 基: 计算机一般都以 2 为基;
  - 尾数的位数: 确定有效数字的位数, 即精度;
  - 指数的位数: 确定所能表示的数的范围.
- 在 IEEE 754 标准中, 浮点数是用二进制表示的, 由三部分组成: 符号 (s), 指数 (e) 和尾数 (f), 见下图.

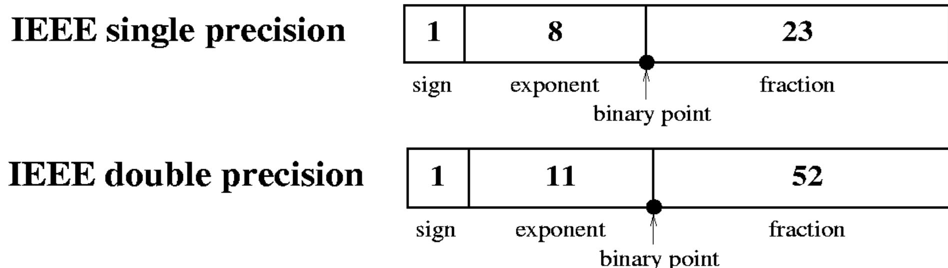


图 1.2 IEEE 754 中单精度格式与双精度格式的位模式

- 单精度格式: 用 8 位字长的二进制数来表示指数, 因此 e 的取值范围为 $[0, 255]$. 当 $0 \leq e < 255$ 时, 按单精度格式存储的数, 其对应的值是使用以下方法得到的:

  将二进制基数点 (小数点) 插入到尾数 f 最高有效位的左侧, 并将一个**隐含位**插入到二进制基数点的左侧, 因而得到的是一个二进制带分数 (整数加小数).

  如此构成的带分数为**单精度格式有效数字**. 隐含位的值并没有显式指定 (不存储), 而是通过指数 e 的值来隐式指定:

  IEEE 754 中规定, 当 $0 < e < 255$ 时, 表示的单精度数为**二进制正规数**, 此时隐含位为 1. 当 $e = 0$ 时, 表示的单精度数为**二进制次正规数**, 隐含位为 0.

  ☞ 这里的正规数与前面的定义有点区别, 因为这里引入了隐含位. 为了以示区别, 我们加上二进制.

  单精度格式位模式中的尾数只有 23 位, 但由于使用了隐含位, 所以能提供 24 位有效数字 (二进制). 在 IEEE 中, 单精度格式与其表示的值的对应关系是

单精度格式位模式	值
$0 < e < 255$	$(-1)^s \times 1.f \times 2^{e-127}$ (二进制正规数)
$e = 0, f \neq 0$	$(-1)^s \times 0.f \times 2^{-126}$ (二进制次正规数)
$e = 0, f = 0$	$(-1)^s \times 0.0$ (有符号的零)
$e = 255, f = 0, s = 0$	+inf (正无穷大)
$e = 255, f = 0, s = 1$	-inf (负无穷大)
$e = 255, f \neq 0$	NaN (非数、非确定值)

  其中 127 是单精度格式的**指数偏移值** (exponent bias), 在 IEEE 标准中, 这个值定义为 $2^{(\text{指数位长}-1)} - 1$. 所以对于单精度格式, 指数偏移值就是 $2^{8-1} - 1 = 127$, 而对于双精度格式, 这个值为 $2^{11-1} - 1 = 1023$.

- 双精度格式: 与单精度格式类似, 对应关系是

双精度格式位模式	值
$0 < e < 2047$	$(-1)^s \times 1.f \times 2^{e-1023}$ (二进制正规数)
$e = 0, f \neq 0$	$(-1)^s \times 0.f \times 2^{-1022}$ (二进制次正规数)
$e = 0, f = 0$	$(-1)^s \times 0.0$ (有符号的零)
$e = 2047, f = 0, s = 0$	+inf (正无穷大)
$e = 2047, f = 0, s = 1$	-inf (负无穷大)
$e = 2047, f \neq 0$	NaN (非数、非确定值)

例 1.11 单精度格式所能表示的十进制数范围.

- 最大二进制正规数为 $7F7FFFFF_{16} = 3.40282347 \times 10^{38}$
- 最小 (正的) 二进制正规数为 $00800000_{16} = 1.17549435 \times 10^{-38}$
- 最大二进制次正规数为 $007FFFFF_{16} = 1.17549421 \times 10^{-38}$
- 最小 (正的) 二进制次正规数为 $00000001_{16} = 1.40129846 \times 10^{-45}$

例 1.12 双精度格式所能表示的十进制数范围.

- 最大二进制正规数为 $7FEFFFFF\ FFFFFFFF_{16} = 1.7976931348623157 \times 10^{308}$
- 最小 (正的) 二进制正规数为 $00100000\ 00000000_{16} = 2.2250738585072014 \times 10^{-308}$
- 最大二进制次正规数为 $000FFFFF\ FFFFFFFF_{16} = 2.2250738585072009 \times 10^{-308}$
- 最小 (正的) 二进制次正规数为 $00000000\ 00000001_{16} = 4.9406564584124654 \times 10^{-324}$
- $3FF00000\ 00000000_{16} = 1$

> 在 Matlab 中, `hex2num` 可以将一个由 16 个 16 进制数组成的字符串转化为其所对应的浮点数 (根据 IEEE 标准), 类似的命令有 `hex2dec, bin2dec, base2dec, num2hex, dec2bin, ...`

例 1.13 把二进制数 $(1001.0101)_2$ 转换成十进制数.

$$\begin{aligned}(1001.0101)_2 &= 1\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0 + 0\times 2^{-1} + 1\times 2^{-2} + 0\times 2^{-3} + 1\times 2^{-2} \\ &= 9.3125_{10}\end{aligned}$$

例 1.14 把十进制数 13.125_{10} 转换成二进制数.

整数部分: $13_{10} = 1101_2$

小数部分:

- $0.125 \times 2 = 0.25$, 整数位是 $0 \to .0$;
- $0.25 \times 2 = 0.5$, 整数位是 $0 \to .00$;
- $0.5 \times 2 = 1$, 整数位是 $1 \to .001$;

所以 $13.125_{10} = 1101.001_2$

一个十进制数能否用二进制浮点数精确表示, 关键在于小数部分.

例 1.15 十进制数 0.1_{10} 能否用二进制数精确表示?

- $0.1 \times 2 = 0.2$, 整数位是 $0 \to .0$;
- $0.2 \times 2 = 0.4$, 整数位是 $0 \to .00$;
- $0.4 \times 2 = 0.8$, 整数位是 $0 \to .000$;
- $0.8 \times 2 = 1.6$, 整数位是 $1 \to .0001$;
- $0.6 \times 2 = 1.2$, 整数位是 $1 \to .00011$;
- $0.2 \times 2 = 0.4$, 整数位是 $0 \to .000110$;
- $0.4 \times 2 = 0.8$, 整数位是 $0 \to .0001100$;
- $0.8 \times 2 = 1.6$, 整数位是 $1 \to .00011001$;
- $0.6 \times 2 = 1.2$, 整数位是 $1 \to .000110011$;
- $\ldots\ldots$

得到一个无限循环的二进制小数, 显然用有限位字长是无法表示的, 因此 0.1_{10} 无法用 IEEE 754 浮点数精确表示.

同理可知 0.2, 0.4, 0.6, 0.8, 0.3, 0.7, 0.9 是无法用二进制数或 IEEE 754 浮点数精确表示的, 故 0.1 至 0.9 的 9 个小数中, 只有 0.5 可以用 IEEE 754 浮点数精确表示.

例 1.16 能用二进制数精确表示的十进制数. 易知

$$
\begin{aligned}
&0.1_2 = 2_{10}^{-1} = 0.5\\
&0.01_2 = 2_{10}^{-2} = 0.25\\
&0.001_2 = 2_{10}^{-3} = 0.125\\
&0.0001_2 = 2_{10}^{-4} = 0.0625\\
&0.00001_2 = 2_{10}^{-5} = 0.03125\\
&0.000001_2 = 2_{10}^{-6} = 0.015625\\
&0.0000001_2 = 2_{10}^{-7} = 0.0078125\\
&0.00000001_2 = 2_{10}^{-8} = 0.00390625\\
&\cdots\cdots
\end{aligned}
$$

由此可知, 一个十进制小数要能用浮点数精确表示, 最后一位必须是 5. 当然这是必要条件, 并非充分条件. 如 0.35 就无法精确表示.

例 1.17 N 位二进制小数能精确表示的非零十进制小数总共有多少个?

- 1 位二进制小数能精确表示的有 $2^0 = 1$ 个 $(0.1_2 = 0.5_{10})$;
- 2 位二进制小数能精确表示的有 $2^1 = 2$ 个 $(0.01_2 = 0.25_{10}, 0.11_2 = 0.75_{10})$;
- 3 位二进制小数能精确表示的有 $2^2 = 4$ 个
- $\cdots\cdots$
- N 位二进制小数能精确表示的有 2^{N-1} 个

所以 N 位二进制小数能精确表示的十进制小数总共有 $2^N - 1$ 个.

1.4.2 IEEE 中的浮点数运算

- IEEE 754 标准也定义了浮点数的运算规则:
  - **加、减、乘、除、平方根、余数、将浮点格式的数舍入为整数值、在不同浮点格式之间转换、在浮点和整数格式之间转换以及比较**. IEEE 对以上浮点运算的准确度作了规定: 求余和比较运算必须精确无误. 其他运算必须向其目标提供精确的结果, 除非没有此类结果, 或者该结果不满足目标格式, 此时运算必须按照下面介绍的舍入模式对精确结果进行最低限度的修改, 并将经过修改的结果提供给运算的目标.
  - 在十进制字符串和两种基本浮点格式之一的二进制浮点数之间进行转换的准确度、单一性和一致性要求.
    对于在指定范围内的操作数, 这些转换必须生成精确的结果 (如果可能的话), 或者按照规定的舍入模式, 对此类精确结果进行最低限度的修改. 对于不在指定范围内的操作数, 这些转换生成的结果与精确结果之间的差值不得超过取决于舍入模式的指定误差.
  - 五种类型的 IEEE 浮点异常, 以及用于向用户指示发生这些类型异常的条件.
    五种类型的浮点异常是: **无效运算、被零除、上溢、下溢和不精确**.
  - IEEE 中有四种舍入模式: 向**最接近的可表示的值** (就近舍入); 当有两个最接近的可表示的值时首选 “偶数” 值 (二进制意义下); 向**负无穷大方向** (向下); 向**正无穷大方向** (向上) 以及向 **0 方向** (截断).

☞ 不同编译器对舍入可能有不同的处理方式.

- 下溢
  当运算结果非常小时, 就会发生下溢. 下表是下溢阈值.

目标的精度	下溢阈值	
单精度	最小正规数	$1.17549435 \times 10^{-38}$
	最大次正规数	$1.17549421 \times 10^{-38}$
双精度	最小正规数	$2.2250738585072014 \times 10^{-308}$
	最大次正规数	$2.2250738585072009 \times 10^{-308}$

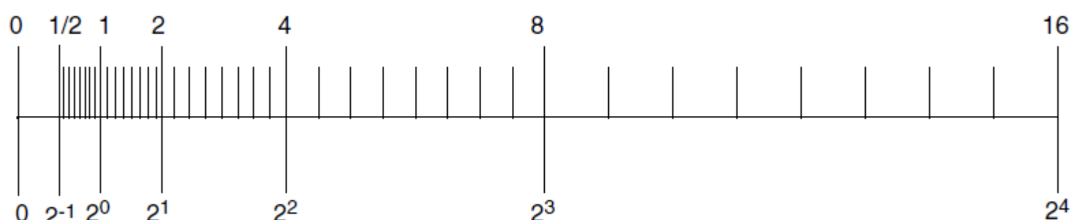
  IEEE 算法处理下溢的方式是**渐进下溢**: 当生成的正确结果的数量级低于最小正规数时, 就会生成次正规数, 而不是返回零.
- **机器精度**: 将 1.0 与大于 1.0 的最小浮点数之间的距离称为 machine epsilon, 它一半称为 unit roundoff, 记为 ε_m, 我们一般也称之为机器精度, 它是计算机表示一个浮点数时的相对误差界, 即 [21, page 38,39]

$$\mathrm{fl}(x) = x(1+\delta) \quad \text{或} \quad \mathrm{fl}(x) = \frac{x}{1+\delta}, \quad |\delta| \le \varepsilon_m.$$

  这里 $\mathrm{fl}(x)$ 表示 x 在计算机中实际存储的 IEEE 浮点数.
  在 IEEE 标准下, 单精度和双精度浮点运算的最大相对误差 ε_m 分别为

精度	最大相对误差
单精度	$2^{-24} \approx 5.960464 \times 10^{-8}$
双精度	$2^{-53} \approx 1.110223 \times 10^{-16}$

例 1.18 假定要使用只有三个精度位的二进制算法. 那么, 最大相对误差为 2^{-3}. 在任意两个 2 的幂之间, 只有 $2^3 - 1 = 7$ 个可表示数字, 如下图所示.



数轴显示了数字之间的差距是随着指数增加而加倍增加的.

在 IEEE 单精度格式中, 两个最小正次正规数之间的差大约是 10^{-45}, 而两个最大有限数之间的数量级差大约是 10^{31}!

- 当运算结果处于两个浮点数的正中间时, 选取最低有效位为 0 的浮点数作为近似值 (“偶数”).
- 精确是偶然的, 误差是必然的. 做数值算法, 惟一能做的就是尽量使误差不积累.

1.4.3 浮点运算误差分析

由于计算机无法精确表示所有的浮点数, 在做浮点运算时, 如果计算结果无法精确表示, 此时就会产生的误差, 即**舍入误差**. IEEE 浮点运算标准中规定, 如果 $a \odot b$ 的结果无法精确表示, 则用一个最接近的浮点数来代替, 记为 $\mathrm{fl}(a \odot b)$. 这里的 $\odot$ 表示加、减、乘、除四种二进制运算符. 如果精确值位于相邻的两个浮点数的正中间, 则取其中最低有效位（二进制表示）为 0 的浮点数作为近似值.

在不考虑溢出的情况下, 我们有

$$\mathrm{fl}(a \odot b) = (a \odot b)(1+\delta) \quad \text{或} \quad \mathrm{fl}(a \odot b) = \frac{a \odot b}{1+\delta}, \tag{1.9}$$

其中 δ 表示浮点运算的相对误差, 满足 $|\delta| \le \varepsilon_m$. 公式 (1.9) 是分析浮点运算舍入误差的基础 [21, page 40]. IEEE 浮点运算标准同时也规定, 对于开根号运算, 产生的误差同样也满足 (1.9).

引理 1.4 [21] 设 $|\delta_i| \le \varepsilon_m$, $n\varepsilon_m < 1$, 则

$$\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = 1+\theta_n, \quad |\theta_n| \le \gamma_n \triangleq \frac{n\varepsilon_m}{1-n\varepsilon_m},$$

其中 $\rho_i = \pm 1$.

例 1.19 (多项式运算的舍入误差分析) : 已知多项式 $p(x) = \sum_{k=0}^{n} a_k x^k$. 对于给定的 x, 分析计算 $p(x)$ 的值时的舍入误差.

解. 在对多项式 $p(x) = a_n x^n + a_{n-1}x^{n-1} \cdots + a_1 x + a_0$ 求值时, 我们通常是基于 Horner 法则, 即

算法 1.1 Horner 法则

```
1: p = a_n
2: for i = n − 1 : −1 : 0 do
3:     p = x * p + a_i
4: end for
```

先看一个具体的例子. 设 $p(x) = (x-2)^9$, 观测 $x = 2$ 附近的舍入误差. 我们通过画图来显示误差情况（见图 1.3）. 通过观察可以发现, 用 Horner 法则求值时会在 $x = 2$ 附近会出现 “噪声带”.

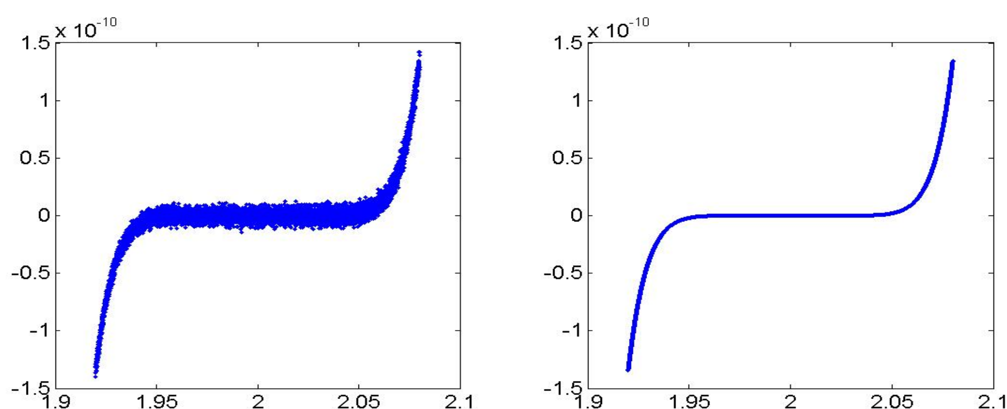


图 1.3 分别利用 Horner 法则（左图）和 $y = (x-2)^9$（右图）在区间 $[1.92, 2.08]$ 上的 8000 个等分点上求值的结果

下面我们基于舍入误差分析模型 (1.9) 来分析多项式求值的误差情况. 带舍入误差的 Horner 算法可写为

算法 1.2 带舍入误差的 Horner 法则

1: $p = a_n$
2: **for** $i = n-1 : -1 : 0$ **do**
3: $\quad p = ((x * p)(1+\delta_i) + a_i)(1+\tilde{\delta}_i)$ % $|\delta_i| \le \varepsilon_m, |\tilde{\delta}_i| \le \varepsilon_m$
4: **end for**

所以得到的最终计算结果为

$$\mathrm{fl}(p(x)) = \sum_{i=0}^{n-1}\left((1+\tilde{\delta}_i)\prod_{j=0}^{i-1}(1+\delta_j)(1+\tilde{\delta}_j)\right)a_i x^i + \left(\prod_{j=0}^{n-1}(1+\delta_j)(1+\tilde{\delta}_j)\right)a_n x^n. \tag{1.10}$$

假定 $j \cdot \varepsilon_m \ll 1$, 则有

$$\begin{aligned}(1+\delta_1)\cdots(1+\delta_j) &\le (1+\varepsilon_m)^j \lesssim 1 + j\cdot\varepsilon_m,\\ (1+\delta_1)\cdots(1+\delta_j) &\ge (1-\varepsilon_m)^j \ge 1 - j\cdot\varepsilon_m.\end{aligned}$$

于是 (1.10) 可改写为

$$\mathrm{fl}(p(x)) = \sum_{i=0}^{n}\left(1+\hat{\delta}_i\right)a_i x^i \triangleq \sum_{i=0}^{n}\tilde{a}_i x^i, \quad \text{其中} \quad |\hat{\delta}_i| \le 2n\cdot\varepsilon_m. \tag{1.11}$$

所以 $\mathrm{fl}(p(x))$ 可看作是另一个多项式的精确值, 且这个多项式的系数是原多项式系数的一个小的扰动. 这说明多项式计算的 Horner 算法是向后稳定的, 且系数的向后相对误差不超过 $2n\cdot\varepsilon_m$. 因此, 绝对误差为

$$\begin{aligned}|\mathrm{fl}(p(x)) - p(x)| &= \left|\sum_{i=0}^{n}\tilde{a}_i x^i - \sum_{i=0}^{n}a_i x^i\right|\\ &= \left|\sum_{i=0}^{n}\hat{\delta}_i a_i x^i\right| \le \sum_{i=0}^{n}\left|\hat{\delta}_i a_i x^i\right| \le 2n\cdot\varepsilon_m\sum_{i=0}^{n}|a_i x^i|.\end{aligned}$$

相对误差为

$$\frac{|\mathrm{fl}(p(x)) - p(x)|}{|p(x)|} \le 2n\cdot\varepsilon_m\frac{\sum_{i=0}^{n}|a_i x^i|}{\left|\sum_{i=0}^{n}a_i x^i\right|}.$$

□

1.5 课后习题

1.1 证明定理 1.29:
设 $(\cdot,\cdot)$ 是 $\mathbb{C}^n$ 上的一个内积, 则存在一个 Hermite 正定矩阵 $A \in \mathbb{C}^{n\times n}$ 使得 $(x,y) = y^*Ax$ 对任意 $x,y \in \mathbb{C}^n$ 都成立. 反之, 若 $A \in \mathbb{C}^{n\times n}$ 是 Hermite 正定矩阵, 则 $f(x,y) \triangleq y^*Ax$ 是 $\mathbb{C}^n$ 上的一个内积.

1.2 证明: $\|x\|_\infty = \lim\limits_{p\to\infty}\left(\sum\limits_{i=1}^{n}|x_i|^p\right)^{\frac{1}{p}}$.

1.3 证明定理 1.7 中的三个不等式:
(1) $\|x\|_2 \le \|x\|_1 \le \sqrt{n}\,\|x\|_2$,
(2) $\|x\|_\infty \le \|x\|_2 \le \sqrt{n}\,\|x\|_\infty$,
(3) $\|x\|_\infty \le \|x\|_1 \le n\,\|x\|_\infty$.

1.4 证明定理 1.8, 即 Cauchy-Schwartz 不等式:
设 $(\cdot,\cdot)$ 是 $\mathbb{C}^n$ 上的内积, 则对任意 $x,y\in\mathbb{C}^n$, 有 $|(x,y)|^2 \le (x,x)\cdot(y,y)$

1.5 证明:
(1) $\|A\|_1 = \max\limits_{1\le j\le n}\left(\sum\limits_{i=1}^{n}|a_{ij}|\right)$; (2) $\|A\|_2 = \max\limits_{x\ne0,\,y\ne0}\dfrac{y^TAx}{\|x\|_2\|y\|_2}$.

1.6 证明:
(1) 对任意的算子范数 $\|\cdot\|$, 有 $\|I\| = 1$;
(2) 对任意的相容范数 $\|\cdot\|$, 有 $\|I\| \ge 1$.

1.7 证明: $\|A\| \triangleq \max\limits_{1\le i,j\le n}|a_{ij}|$ 是矩阵范数, 但不是相容范数.

1.8 证明:
(1) $\dfrac{1}{\sqrt{n}}\|A\|_2 \le \|A\|_1 \le \sqrt{n}\,\|A\|_2$;
(2) $\dfrac{1}{\sqrt{n}}\|A\|_F \le \|A\|_1 \le \|A\|_F$;
(3) $\|A\|_2 \le \|A\|_F \le \sqrt{n}\|A\|_2$;
(4) $\|A\|_2^2 \le \|A\|_1\cdot\|A\|_\infty$.

1.9 设 $A\in\mathbb{R}^{n\times n}$ 是正交矩阵. 证明: $\det(A) = \pm1$.

1.10 设 $A,B\in\mathbb{R}^{n\times n}$ 都是正交矩阵, 且 $\det(A) = -\det(B)$. 证明: $A+B$ 奇异.

1.11 设 $A\in\mathbb{R}^{n\times n}$. 证明: $\mathrm{rank}(A) = 1$ 的充要条件是存在非零向量 $a,b\in\mathbb{R}^n$ 使得 $A = ab^T$.

1.12 设 A_k 是 $A\in\mathbb{R}^{n\times n}$ 的一个 k 阶子矩阵, $\|\cdot\|_p$ 是任意一个算子范数. 证明: $\|A_k\|_p \le \|A\|_p$.

1.13 设 $\|\cdot\|$ 是 $\mathbb{R}^m$ 空间上的一个向量范数, $A\in\mathbb{R}^{m\times n}$, 且 $\mathrm{rank}(A) = n$.
证明: $|||x||| \triangleq \|Ax\|$ 是一个向量范数.

1.14 设 $\|\cdot\|$ 是 $\mathbb{R}^n$ 空间上的一个向量范数. 证明: $\|A^{-1}\|^{-1} = \min\limits_{\|x\|=1}\|Ax\|$.

1.15 设 $A\in\mathbb{R}^{n\times n}$, $x\in\mathbb{R}^n$ 且 $x\ne0$. 证明

$$\left\|A\left(I-\frac{xx^T}{x^Tx}\right)\right\|_F^2 = \|A\|_F^2 - \frac{\|Ax\|_2^2}{x^Tx}.$$

1.16 设 $A, B \in \mathbb{R}^{n\times n}$, 矩阵

$$C = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}, \quad D = \begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix}.$$

证明: $\|C\|_2 = \max\{\|A\|_2, \|B\|_2\}$, $\|D\|_2 = \max\{\|A\|_2, \|B\|_2\}$.

1.17 证明定理 1.26:
$A \in \mathbb{C}^{n\times n}$ 正定 (半正定) 的充要条件是矩阵 $H = (A + A^*)/2$ 正定 (半正定).

1.18 证明定理 1.27:
$A \in \mathbb{R}^{n\times n}$ 正定 (半正定) 的充要条件是对任意非零向量 $x \in \mathbb{R}^n$ 有 $x^T Ax > 0$ ($x^T Ax \geq 0$).

1.19 设 $A \in \mathbb{R}^{n\times n}$ 正定, 证明: A^{-1} 也正定.

1.20 设 $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times m}$, 证明: $\mathrm{tr}(AB) = \mathrm{tr}(BA)$.

1.21 (Sherman-Morrison 公式) 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, $x, y \in \mathbb{R}^n$.
证明: 若 $y^T A^{-1} x \neq 1$, 则 $A - xy^T$ 可逆, 且

$$(A - xy^T)^{-1} = A^{-1} - \frac{A^{-1}xy^T A^{-1}}{y^T A^{-1} x - 1}.$$

1.22 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, $X, Y \in \mathbb{R}^{m\times n}$ $(m \geq n)$.
证明: 若 $Y^T A^{-1} X - I$ 非奇异, 则 $A - XY^T$ 可逆, 且

$$(A - XY^T)^{-1} = A^{-1} - A^{-1}X(Y^T A^{-1} X - I)^{-1} Y^T A^{-1}.$$

(提示：可利用公式 $B^{-1} = A^{-1} - B^{-1}(B - A)A^{-1}$)

1.23 设 $A = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix}$, 其中 $B \in \mathbb{C}^{m\times m}$, $D \in \mathbb{C}^{n\times n}$ 均为上三角矩阵, 且 $B - D$ 非奇异.
证明: 存在矩阵 S, 使得 $S^{-1}AS = \begin{bmatrix} B & 0 \\ 0 & D \end{bmatrix}$. (提示: 可设 $S = \begin{bmatrix} I & \tilde{S} \\ 0 & I \end{bmatrix}$)

1.24 设矩阵 A, B 都是可对角化的, 即 $A = X\Lambda_A X^{-1}$, $B = Y\Lambda_B Y^{-1}$.
试给出 $A \otimes B$ 的特征值分解.

1.25 设 Jordan 块矩阵

$$J = \begin{bmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

试证明:

$$J^k = \begin{bmatrix} \lambda^k & \binom{k}{1}\lambda^{k-1} & \cdots & \binom{k}{n-1}\lambda^{k-(n-1)} \\ & \lambda^k & \cdots & \binom{k}{n-2}\lambda^{k-(n-2)} \\ & & \ddots & \vdots \\ & & & \lambda^k \end{bmatrix},$$

其中 $\binom{k}{p}$ 是二项式系数, 即

$$\binom{k}{p} = \begin{cases} \dfrac{k!}{p!(k-p)!}, & p \le k; \\ 0, & p > k. \end{cases}$$

1.26 考虑抽样数据 $x_1, x_2, \ldots, x_n$, 其均值为 $\bar{x} = \dfrac{1}{n}\sum_{i=1}^{n} x_i$, 样本方差为 (无偏估计)

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2.$$

试证明:

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i^2 - n\bar{x}^2).$$

对于数值计算, 上述两种计算方法哪一个更可靠, 为什么?

1.27 根据浮点运算的舍入误差模型 (1.9), 在不考虑溢出的情况下, 证明 [21]

$$\mathrm{fl}\left(\sum_{i=1}^{n} x_i y_i\right) = \sum_{i=1}^{n} x_i y_i (1+\theta_i), \quad \text{其中 } |\theta_i| \le \gamma_i \triangleq \frac{i\varepsilon_m}{1 - i\varepsilon_m},\ i = 1, 2, \ldots, n.$$

实践题

1.28 数值计算中的误差. 已知 $\sin(x)$ 的幂级数展开为

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

试利用该公式编程计算 $\sin(\pi/2)$ 和 $\sin(33\pi/2)$ 的值, 并与实际值做比较, 误差分别多大? 分析原因.

第二章　线性方程组的直接解法

一般来说, 求解线性方程组的数值方法可以分为两类: 直接法与迭代法. 本章介绍直接法, 即 Gauss 消去法. 直接法相对比较稳定, 因此在工程计算中很受欢迎. 但由于运算量是 $\mathcal{O}(n^3)$, 这里 n 表示未知量的个数, 当问题规模较大时, 时间会很长. 目前, 直接法主要用于小规模或中等规模线性方程组的数值求解.

2.1 Gauss 消去法和 LU 分解

2.1.1 LU 分解

考虑线性方程组

$$Ax = b, \tag{2.1}$$

其中 $A \in \mathbb{R}^{n\times n}$ 非奇异, $b \in \mathbb{R}^n$ 为给定的右端项. Gauss 消去法本质上就是对系数矩阵 A 进行 LU 分解, 即将 A 分解成两个矩阵的乘积

$$A = LU, \tag{2.2}$$

其中 L 是单位下三角矩阵, U 为上三角矩阵. 这个分解就称为 **LU 分解**.

假定矩阵 A 存在 LU 分解 (2.2), 则方程组 (2.1) 就转化为求解下面两个三角方程组

$$\begin{cases} Ly = b, \\ Ux = y. \end{cases}$$

显然, 这两个方程组都非常容易求解.

基于 LU 分解的 Gauss 消去法描述如下:

算法 2.1 Gauss 消去法

1: 将 A 进行 LU 分解: $A = LU$, 其中 L 为单位下三角矩阵, U 为上三角矩阵;
2: 利用向前回代, 求解 $Ly = b$, 即得 $y = L^{-1}b$;
3: 利用向后回代, 求解 $Ux = y$, 即得 $x = U^{-1}y = (LU)^{-1}b = A^{-1}b$.

我们知道, 当系数矩阵 A 非奇异时, 方程组 (2.1) 总是存在唯一解. 但是, 并不是每个非奇异矩阵都存在 LU 分解.

定理 2.1 (LU 分解的存在性和唯一性) 设 $A \in \mathbb{R}^{n\times n}$. 则存在唯一的单位下三角矩阵 L 和非奇异上三角矩阵 U, 使得 $A = LU$ 的充要条件是 A 的所有顺序主子矩阵 $A_k = A(1:k, 1:k)$ 都非奇异, $k = 1, 2, \ldots, n$.

证明. 必要性: 设 A_{11} 是 A 的 k 阶顺序主子矩阵, 将 $A = LU$ 写成分块形式

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} L_{11}U_{11} & L_{11}U_{12} \\ L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} \end{bmatrix}.$$

可得 $A_{11} = L_{11}U_{11}$. 由于 L_{11} 和 U_{11} 均非奇异, 所以 A_{11} 也非奇异.

充分性: 用归纳法.

当 $n = 1$ 时, 结论显然成立.

假设结论对 $n-1$ 阶矩阵都成立, 即对任意 $n-1$ 阶矩阵 $\tilde{A}$, 如果其所有的顺序主子矩阵都非奇异, 则 $\tilde{A}$ 存在 LU 分解.

对任一 n 阶的矩阵 A, 可写成

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

其中 $A_{11} \in \mathbb{R}^{(n-1)\times(n-1)}$ 是 A 的 $n-1$ 阶顺序主子矩阵. 有归纳假设可知, A_{11} 存在 LU 分解, 即存在单位下三角矩阵 L_{11} 和非奇异上三角矩阵 U_{11} 使得

$$A_{11} = L_{11}U_{11}.$$

令

$$L_{21} = A_{21}U_{11}^{-1}, \quad U_{12} = L_{11}^{-1}A_{12}, U_{22} = A_{22} - L_{21}U_{12}$$

则

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11}U_{11} & L_{11}U_{12} \\ L_{21}U_{11} & U_{22} + L_{21}U_{12} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & 1 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \triangleq LU,$$

即 A 存在 LU 分解.

下面证明唯一性. 设 A 存在两个不同的 LU 分解:

$$A = LU = \tilde{L}\tilde{U},$$

其中 L 和 $\tilde{L}$ 为单位下三角矩阵, U 和 $\tilde{U}$ 为非奇异上三角矩阵. 则有

$$L^{-1}\tilde{L} = U\tilde{U}^{-1},$$

该等式左边为下三角矩阵, 右边为上三角矩阵, 所以只能是对角矩阵. 由于单位下三角矩阵的逆仍然是单位下三角矩阵, 所以 $L^{-1}\tilde{L}$ 的对角线元素全是 1, 所以

$$L^{-1}\tilde{L} = I,$$

即 $\tilde{L} = L, \tilde{U} = U$. 由归纳法可知, 结论成立. □

2.1.2 LU 分解的实现过程

给定一个矩阵

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

我们可以通过矩阵初等变换来构造 A 的 LU 分解.

- 第一步: 假定 $a_{11} \neq 0$, 构造矩阵

$$L_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ l_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad 其中 \quad l_{i1} = \frac{a_{i1}}{a_{11}}, i = 2, 3, \ldots, n.$$

易知 L_1 的逆为

$$L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -l_{21} & 1 & 0 & \cdots & 0 \\ -l_{31} & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ -l_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

用 L_1^{-1} 左乘 A, 并将所得到的矩阵记为 $A^{(1)}$, 则

$$A^{(1)} = L_1^{-1}A \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix}.$$

即左乘 L_1^{-1} 后, A 的第一列中除第一个元素外其它都变为 0.

- 第二步: 类似的, 我们可以将上面的操作作用在 $A^{(1)}$ 的子矩阵 $A^{(1)}(2:n, 2:n)$ 上, 将其第一列除第一个元素外都变为 0. 也就是说, 假定 $a_{22}^{(1)} \neq 0$, 构造矩阵

$$L_2 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & l_{n2} & 0 & \cdots & 1 \end{bmatrix}, \quad \text{其中} \quad l_{i2} = \frac{a_{i2}^{(1)}}{a_{22}^{(1)}}, i = 3, 4, \ldots, n.$$

用 L_2^{-1} 左乘 $A^{(1)}$, 并将所得到的矩阵记为 $A^{(2)}$, 则

$$A^{(2)} = L_2^{-1}A = L_2^{-1}L_1^{-1}A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}.$$

- 依此类推, 假定 $a_{kk}^{(k-1)} \neq 0\,(k = 3, 4, \ldots, n-1)$, 则我们可以构造一系列的矩阵 $L_3, L_4, \ldots, L_{n-1}$, 使得

$$L_{n-1}^{-1} \cdots L_2^{-1}L_1^{-1}A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n-1)} \end{bmatrix}$$

为一个上三角矩阵. 我们将这个上三角矩阵记为 U, 并记

$$L = L_1L_2 \cdots L_{n-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{bmatrix},$$

则可得

$$A = LU,$$

这就是 A 的 LU 分解.

将上面的过程写成算法, 描述如下:

算法 2.2 LU 分解

```
1:  for k = 1 to n − 1 do
2:      for i = k + 1 to n do
3:          l_ik = a_ik / a_kk      % 计算 L 的第 k 列
4:      end for
5:      for j = k to n do
6:          u_kj = a_kj      % 计算 U 的第 k 行
7:      end for
8:      for i = k + 1 to n do
9:          for j = i + 1 to n do
10:             a_ij = a_ij − l_ik u_kj      % 更新 A(i+1 : n, i+1 : n)
11:         end for
12:     end for
13: end for
```

Gauss 消去法的运算量

由算法 2.2 可知, LU 分解的运算量 (含加减乘除) 为

$$\sum_{i=1}^{n-1}\left(\sum_{j=i+1}^{n} 1 + \sum_{j=i+1}^{n}\sum_{k=i+1}^{n} 2\right) = \sum_{i=1}^{n-1}\left(n-i+2(n-i)^2\right) = \frac{2}{3}n^3 + O(n^2).$$

由于回代过程的运算量为 $O(n^2)$, 所以 Gauss 消去法的总运算量为 $\frac{2}{3}n^3 + O(n^2)$.

☞ 评价算法的一个标准就是执行时间, 但这依赖于计算机硬件和编程技巧等, 因此直接给出算法执行时间是不太现实的. 所以我们通常是统计算法中算术运算 (加减乘除) 的次数. 在数值算法中, 大多仅仅涉及加减乘除和开方运算. 一般地, 加减运算次数与乘法运算次数具有相同的量级, 而除法运算和开方运算次数具有更低的量级.

☞ 为了尽可能地减少运算量, 在实际计算中, 数, 向量和矩阵做乘法运算时的先后执行次序为: 先计算数与向量的乘法, 然后计算矩阵与向量的乘法, 最后才计算矩阵与矩阵的乘法.

矩阵 L 和 U 的存储

当 A 的第 i 列被用于计算 L 的第 i 列后, 在后面的计算中不再被使用. 同样地, A 的第 i 行被用于计算 U 的第 i 行后, 在后面的计算中也不再被使用. 因此, 为了节省存储空间, 我们可以在计算过程中将 L 的第 i 列存放在 A 的第 i 列, 将 U 的第 i 行存放在 A 的第 i 行, 这样就不需要另外分配空间存储 L 和 U. 计算结束后, A 的上三角部分为 U, 其绝对下三角部分为 L 的绝对下三角部分 (L 的对角线全部为 1, 不需要存储). 此时算法可以描述为:

算法 2.3 LU 分解

1: **for** $k = 1$ to $n - 1$ **do**
2:     **for** $i = k + 1$ to n **do**
3:         $a_{ik} = a_{ik}/a_{kk}$
4:         **for** $j = i + 1$ to n **do**
5:             $a_{ij} = a_{ij} - a_{ik}a_{kj}$
6:         **end for**
7:     **end for**
8: **end for**

MATLAB 源代码 2.1 LU 分解的 Matlab 代码

```
% Matlab code : LU 分解
function A = mylu(A)
n=size(A,1);
for k=1:n-1
  if A(k,k) == 0
    fprintf('Error: A(%d,%d)=0!\n', k, k);
    return;
  end
  for i=k+1:n
    A(i,k)=A(i,k)/A(k,k);
  end
  for i=k+1:n
    for j=i+1:n
      A(i,j)=A(i,j)-A(i,k)*A(k,j);
    end
  end
end
```

☞ 为了充分利用 Matlab 的向量运算优势, 提高运算效率, 上面的程序可改写为

MATLAB 源代码 2.2 LU 分解

```
function A = mylu(A)
n=size(A,1);
for k=1:n-1
  if A(k,k) == 0
    fprintf('Error: A(%d,%d)=0!\n', k, k);
    return;
  end
  A(k+1:n,k)=A(k+1:n,k)/A(k,k);
  A(k+1:n,k+1:n)=A(k+1:n,k+1:n)-A(k+1:n,k)*A(k,k+1:n);
end
```

2.1.3 待定系数法计算 LU 分解

我们也可以利用待定系数法来实现矩阵的 LU 分解. 假设 A 存在 LU 分解, 即 $A = LU$, 或

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & & & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & & & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & u_{33} & \cdots & u_{2n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{bmatrix}.$$

通过比较等式两边的元素来计算 L 和 U 中的各元素的值. 具体计算过程如下:

(1) 比较等式两边的第一行, 可得

$$u_{1j} = a_{1j}, \quad j = 1, 2, \ldots, n.$$

再比较等式两边的第一列, 可得

$$a_{i1} = l_{i1} u_{11} \Rightarrow l_{i1} = a_{i1}/u_{11}, \quad i = 2, 3, \ldots, n.$$

(2) 比较等式两边的第二行, 可得

$$a_{2j} = l_{21} u_{1j} \Rightarrow u_{2j} = a_{2j} - l_{21} u_{ij}, \quad j = 2, 3, \ldots, n.$$

再比较等式两边的第二列, 可得

$$a_{i2} = l_{i1} u_{12} + l_{i2} u_{22} \Rightarrow l_{i1} = (a_{i2} - l_{i1} u_{12})/u_{22}, \quad i = 3, 4, \ldots, n.$$

(3) 以此类推, 第 k 步时, 比较等式两边的第 k 行, 可得

$$u_{kj} = a_{kj} - (l_{k1} u_{1j} + \cdots + l_{k,k-1} u_{k-1,j}), \quad j = k, k+1, \ldots, n.$$

比较等式两边的第 k 列, 可得

$$l_{ik} = (a_{ik} - l_{i1} u_{1k} - \cdots - l_{i,k-1} u_{k-1,k})/u_{kk}, \quad i = k+1, k+2, \ldots, n.$$

直到第 n 步, 即可计算出 L 和 U 的所有元素.

同样, 我们可以利用 A 来存储 L 和 U. 算法描述如下:

算法 2.4 LU 分解 (待定系数法或 Doolittle 方法)

1: **for** $k = 1$ to n **do**

2: $\quad a_{kj} = a_{kj} - \sum_{i=1}^{k-1} a_{ki} a_{ij}, \quad j = k, k+1, \ldots, n$

3: $\quad a_{ik} = \dfrac{1}{a_{kk}} \left(a_{ik} - \sum_{j=1}^{k-1} a_{ij} a_{jk} \right), \quad i = k+1, k+2, \ldots, n$

4: **end for**

相应的 Matlab 程序为:

MATLAB 源代码 2.3 待定系数法 LU 分解

```
function A = mylu2(A)
[n,n]=size(A);
for k=1:n
  A(k,k)=A(k,k)-A(k,1:k-1)*A(1:k-1,k);
  if (A(k,k)==0)
    fprintf('Error: A(%d,%d)=0!\n', i,i); return;
  end
  A(k,k+1:n)=A(k,k+1:n)-A(k,1:k-1)*A(1:k-1,k+1:n);
  A(k+1:n,k)=(A(k+1:n,k)-A(k+1:n,1:k-1)*A(1:k-1,k))/A(k,k);
end
```

2.1.4 三角方程求解

得到 A 的 LU 分解后, 我们最后需要用回代法求解两个三角方程组

$$Ly = b, \quad Ux = y.$$

算法 2.5 向前回代求解 $Ly = b$

1: $y_1 = b_1$
2: **for** $i = 2 : n$ **do**
3: $\quad s = b_i$
4: $\quad$ **for** $j = 1 : i-1$ **do**
5: $\quad\quad s = s - u_{ij}x_j$
6: $\quad$ **end for**
7: $\quad x_i = s$
8: **end for**

算法 2.6 向后回代求解 $Ux = y$

1: $x_n = y_n/u_{nn}$
2: **for** $i = n-1 : -1 : 1$ **do**
3: $\quad s = y_i$
4: $\quad$ **for** $j = i+1 : n$ **do**
5: $\quad\quad s = s - u_{ij}x_j$
6: $\quad$ **end for**
7: $\quad x_i = s/u_{ii}$
8: **end for**

这两个算法的运算量均为 $n^2 + \mathcal{O}(n)$.

2.1.5 选主元 LU 分解

在 LU 分解算法 2.2 中, 我们称 $a_{kk}^{(k-1)}$ 为主元. 如果 $a_{kk}^{(k-1)} = 0$, 则算法就无法进行下去. 即使 $a_{kk}^{(k-1)}$ 不为零, 但如果 $|a_{kk}^{(k-1)}|$ 的值很小, 由于舍入误差的原因, 也可能会给计算结果带来很大的误差. 此时我们就需要通过选主元来解决这个问题.

例 2.1 用 LU 分解求解线性方程组 $Ax = b$, 其中 $A = \begin{bmatrix} 0.02 & 61.3 \\ 3.43 & -8.5 \end{bmatrix}$, $b = \begin{bmatrix} 61.5 \\ 25.8 \end{bmatrix}$, 要求在运算过程中保留 3 位有效数字.

解. 根据 LU 分解算法 2.2, 我们可得

$$\begin{aligned}
&l_{11} = 1.00,\ l_{21} = a_{21}/a_{11} = 1.72 \times 10^2,\ l_{22} = 1.00, \\
&u_{11} = a_{11} = 2.00 \times 10^{-2},\ u_{12} = a_{12} = 6.13 \times 10, \\
&u_{22} = a_{22} - l_{21}u_{12} \approx -8.5 - 1.05 \times 10^4 \approx -1.05 \times 10^4,
\end{aligned}$$

即

$$A \approx \begin{bmatrix} 1.00 & 0 \\ 1.72 \times 10^2 & 1.00 \end{bmatrix} \begin{bmatrix} 2.00 \times 10^{-2} & 6.12 \times 10 \\ 0 & -1.05 \times 10^4 \end{bmatrix}.$$

解方程组 $Ly = b$ 可得

$$y_1 = 6.15 \times 10, \quad y_2 = b_2 - l_{21}y_1 \approx -1.06 \times 10^4.$$

解方程组 $Ux = y$ 可得

$$x_2 = y_2/u_{22} \approx 1.01, \quad x_1 = (y_1 - u_{12} * x_2)/u_{11} \approx -0.413/u_{11} \approx -20.7$$

□

易知, 方程的精确解为 $x_1 = 10.0$ 和 $x_2 = 1.00$. 我们发现 x_1 的误差非常大. 导致这个问题的原因就是 $|a_{11}|$ 太小, 用它做主元时会放大舍入误差. 所以我们需要通过置换矩阵来选主元.

首先介绍置换矩阵的一些基本性质.

引理 2.1 设 $P \in \mathbb{R}^{n\times n}$ 为置换矩阵, $X \in \mathbb{R}^{n\times n}$ 为任意矩阵, 则

(1) PX 相当于将 X 的行进行置换; XP 相当于将 X 的列进行置换;

(2) $P^{-1} = P^T$, 即 P 是正交矩阵;

(3) $\det(P) = \pm 1$;

(4) 置换矩阵的乘积仍然是置换矩阵.

定理 2.2 (选主元 LU 分解的存在性) 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, 则存在置换矩阵 P_1, P_2, 以及单位下三角矩阵 L 和非奇异上三角矩阵 U, 使得 $P_1AP_2 = LU$. 其中 P_1 和 P_2 中只有一个是必需的.

证明. 用归纳法.

当 $n = 1$ 时, 取 $P_1 = P_2 = L = 1, U = A$ 即可.

假设结论对 $n - 1$ 成立.

设 $A \in \mathbb{R}^{n\times n}$ 是 n 阶非奇异矩阵, 则 A 至少存在一个非零元, 取置换矩阵 $\hat{P}_1$ 和 $\hat{P}_2$ 使得

$$\hat{P}_1 A \hat{P}_2 = \begin{bmatrix} a_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

其中 $a_{11}\neq 0$, $A_{22}\in\mathbb{R}^{(n-1)\times(n-1)}$.

$$u_{11}=a_{11},\quad U_{12}=A_{12},\quad L_{21}=A_{21}/a_{11},\quad U_{22}=A_{22}-L_{21}U_{12}.$$

则有

$$\begin{bmatrix}1 & 0\\ L_{21} & I\end{bmatrix}\begin{bmatrix}u_{11} & U_{12}\\ 0 & U_{22}\end{bmatrix}=\begin{bmatrix}a_{11} & A_{12}\\ A_{21} & A_{22}\end{bmatrix}=\hat{P}_1A\hat{P}_2.$$

两边取行列式可得

$$0\neq\det(P_1A\hat{P}_2)=\det\left(\begin{bmatrix}1 & 0\\ L_{21} & I\end{bmatrix}\right)\cdot\det\left(\begin{bmatrix}u_{11} & U_{12}\\ 0 & U_{22}\end{bmatrix}\right)=a_{11}\det(U_{22}).$$

所以 $\det(U_{22})\neq 0$, 即 $U_{22}\in\mathbb{R}^{(n-1)\times(n-1)}$ 非奇异. 由归纳假设可知, 存在置换矩阵 $\tilde{P}_1$ 和 $\tilde{P}_2$ 使得

$$\tilde{P}_1U_{22}\tilde{P}_2=\tilde{L}_{22}\tilde{U}_{22},$$

其中 $\tilde{L}_{22}$ 为单位下三角矩阵, $\tilde{U}_{22}$ 为非奇异上三角矩阵. 取

$$P_1=\begin{bmatrix}1 & 0\\ 0 & \tilde{P}_1\end{bmatrix}\hat{P}_1,\quad P_2=\hat{P}_2\begin{bmatrix}1 & 0\\ 0 & \tilde{P}_2\end{bmatrix},$$

则有

$$\begin{aligned}
P_1AP_2&=\begin{bmatrix}1 & 0\\ 0 & \tilde{P}_1\end{bmatrix}\begin{bmatrix}1 & 0\\ L_{21} & I\end{bmatrix}\begin{bmatrix}u_{11} & U_{12}\\ 0 & U_{22}\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & \tilde{P}_2\end{bmatrix}\\
&=\begin{bmatrix}1 & 0\\ \tilde{P}_1L_{21} & \tilde{P}_1\end{bmatrix}\begin{bmatrix}u_{11} & U_{12}\\ 0 & \tilde{P}_1^T\tilde{L}_{22}\tilde{U}_{22}\tilde{P}_2^T\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & \tilde{P}_2\end{bmatrix}\\
&=\begin{bmatrix}1 & 0\\ \tilde{P}_1L_{21} & \tilde{P}_1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & \tilde{P}_1^T\tilde{L}_{22}\end{bmatrix}\begin{bmatrix}u_{11} & U_{12}\\ 0 & \tilde{U}_{22}\tilde{P}_2^T\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & \tilde{P}_2\end{bmatrix}\\
&=\begin{bmatrix}1 & 0\\ \tilde{P}_1L_{21} & \tilde{L}_{22}\end{bmatrix}\begin{bmatrix}u_{11} & U_{12}\tilde{P}_2\\ 0 & \tilde{U}_{22}\end{bmatrix}\\
&\triangleq LU,
\end{aligned}$$

其中 L 为单位下三角矩阵, U 为非奇异上三角矩阵.

由归纳法可知, 结论成立. □

☞ **第 k 步时, 如何选取置换矩阵 P_1 和 P_2?**

1. 选取 P_1 和 P_2 使得主元为剩下的矩阵中绝对值最大, 这种选取方法称为 "全主元 Gauss 消去法", 简称 GECP (Gaussian elimination with complete pivoting);
2. 选取 P_1 和 P_2 使得主元为第 k 列中第 k 到第 n 个元素中, 绝对值最大, 这种选取方法称为 "部分选主元 Gauss 消去法", 简称 GEPP (Gaussian elimination with partial pivoting), 此时 $P_2=I$, 因此也称为列主元 Gauss 消去法.

- GECP 比 GEPP 更稳定, 但工作量太大, 在实际应用中通常使用 GEPP 算法.
- GEPP 算法能保证 L 所有的元素的绝对值都不超过 1.

算法 2.7 部分选主元 LU 分解

1: $p = 1 : n$;   % 用于记录置换矩阵
2: **for** $i = 1$ to $n - 1$ **do**
3:   $a_{ki} = \max_{i \le j \le n} |a_{ji}|$ % 选列主元
4:   **if** $k \sim= i$ **then**
5:     **for** $j = 1$ to n **do**
6:       $t = a_{ij}$
7:       $a_{ij} = a_{kj}$
8:       $a_{kj} = t$ % 交换 A 的第 i 行与第 k 行
9:     **end for**
10:     $p(k) = i$
11:     $p(i) = k$ % 更新置换矩阵
12:   **end if**
13:   **for** $j = i + 1$ to n **do**
14:     $a_{ji} = a_{ji}/a_{ii}$ % 计算 L 的第 i 列
15:   **end for**
16:   **for** $j = i + 1$ to n **do**
17:     **for** $k = i + 1$ to n **do**
18:       $a_{jk} = a_{jk} - a_{ji} * a_{ik}$ % 更新 $A(i+1:n, i+1:n)$
19:     **end for**
20:   **end for**
21: **end for**

相应的 Matlab 程序如下:

MATLAB **源代码** 2.4 部分选主元 LU 分解

```
1  function [A,p] = myplu(A)
2  [n,n]=size(A);
3  p=1:n;
4  for i=1:n-1
5     [a,k]=max(abs(A(i:n,i)));
6     if a==0
7        error('Error: 第 %d 步的列主元为 0!\n', i);
8     end
9     k=k+i-1;
10    if k~=i
11       Atmp=A(i,:); A(i,:)=A(k,:); A(k,:)=Atmp;
12       ptmp=p(i); p(i)=p(k); p(k)=ptmp;
13    end
14    A(i+1:n,i)=A(i+1:n,i)/A(i,i);
15    A(i+1:n,i+1:n)=A(i+1:n,i+1:n)-A(i+1:n,i)*A(i,i+1:n);
16 end
```

例 2.2 用部分选主元 LU 分解求解线性方程组 $Ax = b$, 其中 $A = \begin{bmatrix} 0.02 & 61.3 \\ 3.43 & -8.5 \end{bmatrix}$, $b = \begin{bmatrix} 61.5 \\ 25.8 \end{bmatrix}$, 要求

在运算过程中保留 3 位有效数字.

解. 由于 $|a_{21}| > |a_{11}|$, 根据部分选主元 LU 分解算法, 我们需要将第一行与第二行交换, 即取 $P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, 然后计算 $\tilde{A} = PA$ 的 LU 分解, 可得

$$
\begin{aligned}
& l_{11} = 1.00, \ l_{21} = \tilde{a}_{21}/\tilde{a}_{11} = 5.83 \times 10^{-3}, \ l_{22} = 1.00, \\
& u_{11} = \tilde{a}_{11} = 3.43, \ u_{12} = \tilde{a}_{12} = -8.50, \\
& u_{22} = \tilde{a}_{22} - l_{21}u_{12} \approx 6.13 \times 10 + 4.96 \times 10^{-2} \approx 6.13 \times 10,
\end{aligned}
$$

即

$$
PA \approx \begin{bmatrix} 1.00 & 0 \\ 5.83 \times 10^{-3} & 1.00 \end{bmatrix} \begin{bmatrix} 3.43 & -8.50 \\ 0 & 6.13 \times 10 \end{bmatrix}.
$$

解方程组 $Ly = P^T b$ 可得

$$
y_1 = 2.58 \times 10, \quad y_2 \approx 6.12 \times 10.
$$

解方程组 $Ux = y$ 可得

$$
x_2 = y_2/u_{22} \approx 0.998, \quad x_1 = (y_1 - u_{12} * x_2)/u_{11} \approx 34.3/u_{11} \approx 10.0
$$

所以, 数值解具有 3 位有效数字. □

2.1.6 矩阵求逆

我们可以通过部分选主元 LU 分解来计算矩阵的逆. 设 $PA = LU$, 则

$$
A^{-1} = P^T U^{-1} L^{-1},
$$

等价于求解下面 $2n$ 个三角线性方程组

$$
Ly_i = Pe_i, \quad Ux_i = y_i, \quad i = 1, 2, \ldots, n.
$$

2.2 特殊方程组的求解

2.2.1 对称正定线性方程组

考虑线性方程组

$$Ax = b$$

其中 $A \in \mathbb{R}^{n\times n}$ 对称正定的.

我们首先给出对称正定矩阵的几个基本性质.

定理 2.3 设 $A \in \mathbb{R}^{n\times n}$.

- A 对称正定当且仅当 A 对称且所有特征值都是正的;
- A 对称正定当且仅当 X^TAX 对称正定, 其中 $X \in \mathbb{R}^{n\times n}$ 是一个任意的非奇异矩阵;
- 若 A 对称正定, 则 A 的任意主子矩阵都对称正定;
- 若 A 对称正定, 则 A 的所有对角线元素都是正的, 且 $\max_{i\neq j}\{|a_{ij}|\} < \max_{i}\{a_{ii}\}$, 即绝对值最大的元素出现在对角线上.

定理 2.4 (Cholesky 分解) 设 $A \in \mathbb{R}^{n\times n}$ 对称正定, 则存在唯一的对角线元素为正的下三角矩阵 L, 使得

$$A = LL^T.$$

该分解称为 Cholesky 分解.

证明. 首先证明存在性, 我们用数学归纳法来构造矩阵 L.

当 $n = 1$ 时, 由 A 的对称正定性可知 $a_{11} > 0$. 取 $l_{11} = \sqrt{a_{11}}$ 即可.

假定结论对所有不超过 $n-1$ 阶的对称正定矩阵都成立. 设 $A \in \mathbb{R}^{n\times n}$ 是 n 阶对称正定, 则 A 可分解为

$$A = \begin{bmatrix} a_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^T & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^T & I \end{bmatrix}^T,$$

其中 $\tilde{A}_{22} = A_{22} - A_{12}^T A_{12}/a_{11}$. 由定理 2.3 可知, $\begin{bmatrix} 1 & 0 \\ 0 & \tilde{A}_{22} \end{bmatrix}$ 对称正定, 故 $\tilde{A}_{22}$ 是 $n-1$ 阶对称正定矩阵. 根据归纳假设, 存在唯一的对角线元素为正的下三角矩阵 $\tilde{L}$, 使得 $\tilde{A}_{22} = \tilde{L}\tilde{L}^T$. 令

$$L = \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^T & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{L} \end{bmatrix} = \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^T & \tilde{L} \end{bmatrix}.$$

易知, L 是对角线元素均为正的下三角矩阵, 且

$$LL^T = \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^T & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{L} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{L}^T \end{bmatrix} \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^T & I \end{bmatrix}^T = A.$$

由归纳法可知, 对任意对称正定实矩阵 A, 都存在一个对角线元素为正的下三角矩阵 L, 使得

$$A = LL^T.$$

唯一性可以采用反证法, 留做作业. □

Cholesky 分解的实现

设 $A = LL^T$, 即

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ & l_{22} & \cdots & l_{n2} \\ & & \ddots & \vdots \\ & & & l_{nn} \end{bmatrix}.$$

直接比较等式两边的元素可得

$$a_{ij} = \sum_{k=1}^{n} l_{ik} l_{jk} = l_{jj} l_{ij} + \sum_{k=1}^{j-1} l_{ik} l_{jk}, \quad i, j = 1, 2, \ldots, n.$$

根据这个计算公式即可得下面的算法:

算法 2.8 Cholesky 分解算法

1: **for** $j = 1$ to n **do**
2: $\quad l_{jj} = \left(a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2\right)^{1/2}$
3: $\quad$ **for** $i = j+1$ to n **do**
4: $\qquad l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}) / l_{jj}$
5: $\quad$ **end for**
6: **end for**

关于 Cholesky 算法的几点说明

- 与 LU 分解一样, 可以利用 A 的下三角部分来存储 L;
- Cholesky 分解算法的运算量为 $\frac{1}{3}n^3 + \mathcal{O}(n^2)$, 大约为 LU 分解的一半;
- Cholesky 分解算法是稳定的（稳定性与全主元 Gauss 消去法相当）, 故不需要选主元.

改进的 Cholesky 分解算法

为了避免开方运算, 我们可以将 A 分解为: $A = LDL^T$, 即

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \begin{bmatrix} 1 & l_{21} & \cdots & l_{n1} \\ & 1 & \cdots & l_{n2} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}.$$

通过待定系数法可得

$$a_{ij} = \sum_{k=1}^{n} l_{ik} d_k l_{jk} = d_j l_{ij} + \sum_{k=1}^{j-1} l_{ik} d_k l_{jk}, \quad i, j = 1, 2, \ldots, n.$$

基于以上分解来求解对称正定线性方程组的算法称为**改进的平方根法**:

算法 2.9 改进的平方根法

```
1:  % 先计算分解
2:  for j = 1 to n do
3:      d_j = a_jj − Σ_{k=1}^{j−1} l_jk^2 d_k
4:      for i = j + 1 to n do
5:          l_ij = (a_ij − Σ_{k=1}^{j−1} l_ik d_k l_jk)/d_j
6:      end for
7:  end for
8:  % 解方程组: Ly = b 和 DL^T x = y
9:  y_1 = b_1
10: for i = 2 to n do
11:     y_i = b_i − Σ_{k=1}^{i−1} l_ik y_k
12: end for
13: x_n = y_n/d_n
14: for i = n − 1 to 1 do
15:     x_i = y_i/d_i − Σ_{k=i+1}^{n} l_ki x_k
16: end for
```

2.2.2 对称不定线性方程组

设 $A \in \mathbb{R}^{n\times n}$ 是非奇异的对称不定矩阵. 若 A 存在 LU 分解, 即 $A = LU$, 则可写成

$$A = LDL^T,$$

其中 D 是由 U 的对角线元素构成的对角矩阵. 然而, 当 A 不定时, 其 LU 分解不一定存在. 若采用选主元 LU 分解, 则其对称性将被破坏. 为了保持对称性, 在选主元时必须对行列进行同样的置换, 即选取置换矩阵 P, 使得

$$PAP^T = LDL^T. \tag{2.3}$$

通常称 (2.3) 为对称矩阵的 LDL^T 分解. 不幸的是, 这样的置换矩阵可能不一定存在, 即分解 (2.3) 不一定存在.

例 2.3 设对称矩阵

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

由于 A 的对角线元素都是 0, 对任意置换矩阵 P, 矩阵 PAP^T 的对角线元素仍然都是 0. 因此, 矩阵 A 不存在分解 (2.3).

基于以上原因, Aasen[1] 提出了下面的分解

$$PAP^T = LTL^T, \tag{2.4}$$

其中 P 为置换矩阵, L 为单位下三角矩阵, T 为对称三对角矩阵. 分解 (2.4) 本质上与部分选主元 LU 分解是一样的,

具体实施细节可参见 [52].

☞ 事实上, 也可以考虑 2×2 块主元, 使得

$$PAP^T = L\tilde{D}L^T,$$

其中 $\tilde{D}$ 是拟对角矩阵, 即块对角矩阵且对角块的大小为 1 或 2. 目前比较广泛使用的块主元策略为 Bunch–Kaufman 策略, 详情可参见 [7].

2.2.3 三对角线性方程组

考虑三对角线性方程组 $Ax=f$, 其中 A 是三对角矩阵:

$$A = \begin{bmatrix} b_1 & c_1 & & \\ a_2 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_n & b_n \end{bmatrix}.$$

我们假定

$$|b_1| > |c_1| > 0, \quad |b_n| > |a_n| > 0, \tag{2.5}$$

且

$$|b_i| \geq |a_i| + |c_i|, \quad a_i c_i \neq 0, \quad i = 2, \ldots, n-1. \tag{2.6}$$

即 A 是不可约弱对角占优的. 此时, 我们可以得到下面的三角分解

$$A = \begin{bmatrix} b_1 & c_1 & & \\ a_2 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_n & b_n \end{bmatrix} = \begin{bmatrix} \alpha_1 & & & \\ a_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & a_n & \alpha_n \end{bmatrix} \begin{bmatrix} 1 & \beta_1 & & \\ & 1 & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & 1 \end{bmatrix} \triangleq LU. \tag{2.7}$$

由待定系数法, 我们可以得到递推公式:

$$\begin{aligned} &\alpha_1 = b_1, \\ &\beta_1 = c_1/\alpha_1 = c_1/b_1, \\ &\begin{cases} \alpha_i = b_i - a_i\beta_{i-1}, \\ \beta_i = c_i/\alpha_i = c_i/(b_i - a_i\beta_{i-1}), \quad i = 2, 3, \ldots, n-1 \end{cases} \\ &\alpha_n = b_n - a_n\beta_{n-1}. \end{aligned}$$

为了使得算法能够顺利进行下去, 我们需要证明 $\alpha_i \neq 0$.

定理 2.5 设三对角矩阵 A 满足条件 (2.5) 和 (2.6). 则 A 非奇异, 且

(1) $|\alpha_1| = |b_1| > 0$;

(2) $0 < |\beta_i| < 1$, $i = 1, 2, \ldots, n-1$;

(3) $0 < |c_i| \leq |b_i| - |a_i| < |\alpha_i| < |b_i| + |a_i|$, $i = 2, 3, \ldots, n$;

证明. 由于 A 是不可约且弱对角占优, 所以 A 非奇异. (见定理 6.8)

结论 (1) 是显然的.

下面我们证明结论 (2) 和 (3).

由于 $0 < |c_1| < |b_1|$, 且 $\beta_1 = c_1/b_1$, 所以 $0 < |\beta_1| < 1$. 又 $\alpha_2 = b_2 - a_2\beta_1$, 所以

$$|\alpha_2| \geq |b_2| - |a_2| \cdot \|\beta_1\| > |b_2| - |a_2| \geq |c_2| > 0, \tag{2.8}$$

$$|\alpha_2| \leq |b_2| + |a_2| \cdot \|\beta_1\| < |b_2| + |a_2|. \tag{2.9}$$

再由结论 (2.8) 和 β_2 的计算公式可知 $0 < |\beta_2| < 1$. 类似于 (2.8) 和 (2.8), 我们可以得到

$$|\alpha_3| \geq |b_3| - |a_3| \cdot \|\beta_2\| > |b_3| - |a_3| \geq |c_3| > 0,$$

$$|\alpha_3| \leq |b_3| + |a_3| \cdot \|\beta_2\| < |b_3| + |a_3|.$$

依此类推, 我们就可以证明结论 (2) 和 (3). □

由定理 2.5 可知, 分解 (2.7) 是存在的. 因此, 原方程就转化为求解 $Ly = f$ 和 $Ux = y$. 由此便可得求解三对角线性方程组的**追赶法**, 其运算量大约为 $8n - 6$.

算法 2.10 追赶法

1: $\beta_1 = c_1/b_1$
2: $y_1 = f_1/b_1$
3: **for** $i = 2$ to $n-1$ **do**
4: $\quad \alpha_i = b_i - a_i\beta_{i-1}$
5: $\quad \beta_i = c_i/\alpha_i$
6: $\quad y_i = (f_i - a_i y_{i-1})/\alpha_i$
7: **end for**
8: $\alpha_n = b_n - a_n\beta_{n-1}$
9: $y_n = (f_n - a_n y_{n-1})/\alpha_n$
10: $x_n = y_n$
11: **for** $i = n-1$ to 1 **do**
12: $\quad x_i = y_i - \beta_i x_{i+1}$
13: **end for**

☞ 具体计算时, 由于求解 $Ly = f$ 与矩阵 LU 分解是同时进行的, 因此, α_i 可以不用存储. 但 β_i 需要存储.

☞ 由于 $|\beta_i| < 1$, 因此在回代求解 x_i 时, 误差可以得到有效控制.

需要指出的是, 我们也可以考虑下面的分解

$$A = \begin{bmatrix} b_1 & c_1 & & \\ a_2 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_n & b_n \end{bmatrix} = \begin{bmatrix} 1 & & & \\ \gamma_2 & 1 & & \\ & \ddots & \ddots & \\ & & \gamma_n & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 & c_1 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & c_{n-1} \\ & & & \alpha_n \end{bmatrix}. \tag{2.10}$$

但此时 $|\gamma_i|$ 可能大于 1. 比如 $\gamma_2 = a_2/b_1$, 因此当 $|b_1| < |a_2|$ 时, $|\gamma_2| > 1$. 所以在回代求解时, 误差可能得不到有效控制. 另外一方面, 计算 γ_i 时也可能会产生较大的舍入误差 (大数除以小数). 但如果 A 是列对角占优, 则可以保证 $|\gamma_i| < 1$.

☞ 如果 A 是 (行) 对角占优, 则采用分解 (2.7); 如果 A 是列对角占优, 则采用分解 (2.10).

2.2.4 带状线性方程组

设 $A \in \mathbb{R}^{n\times n}$ 是带状矩阵, 其下带宽为 b_L, 上带宽为 b_U, 即

$$a_{ij} = 0 \quad \text{for} \quad i > j + b_L \;\text{ or }\; i < j - b_U.$$

其形状如下图所示:

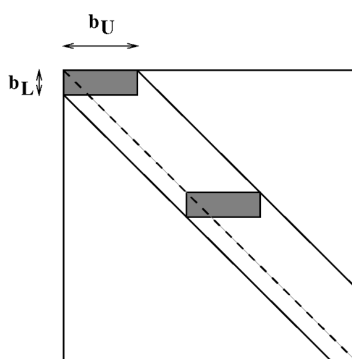


图 2.1 带状矩阵

对于带状矩阵, 其 LU 分解有如下性质:

定理 2.6 设 $A \in \mathbb{R}^{n\times n}$ 是带状矩阵, 其下带宽为 b_L, 上带宽为 b_U. 若 $A = LU$ 是不选主元的 LU 分解, 则 L 为下带宽为 b_L 的带状矩阵, U 为上带宽为 b_U 的带状矩阵. 求解 $Ax = b$ 的运算量大约为 $2nb_Lb_U + 2n(b_L + b_U)$.

若采用部分选主元的 LU 分解, 则有

定理 2.7 设 $A \in \mathbb{R}^{n\times n}$ 是带状矩阵, 其下带宽为 b_L, 上带宽为 b_U. 若 $PA = LU$ 是部分选主元的 LU 分解, 则 U 为上带宽不超过 $b_L + b_U$ 的带状矩阵, L 为下带宽为 b_L 的 "基本带状矩阵", 即 L 每列的非零元素不超过 $b_L + 1$ 个.

2.2.5 Toeplitz 线性方程组

设 $T_n \in \mathbb{R}^{n\times n}$ 是 Toeplitz 矩阵, 即

$$T_n = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+1} \\ t_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix}.$$

易知 Toeplitz 矩阵是反向对称（persymmetric）矩阵, 即关于东北–西南对角线对称. 记 J_n 为 n 阶反向单位矩阵, 即

$$J_n = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \iddots & & \\ 1 & & & \end{bmatrix}.$$

易知 $J_n^T = J_n^{-1} = J_n$.

引理 2.2 矩阵 $A \in \mathbb{R}^{n\times n}$ 是反向对称矩阵当且仅当

$$A = J_n A^T J_n \quad 或 \quad J_n A = A^T J_n.$$

若 A 可逆, 则可得

$$A^{-1} = J_n^{-1} \left(A^T\right)^{-1} J_n^{-1} = J_n \left(A^{-1}\right)^T J_n,$$

即反向对称矩阵的逆也是反向对称矩阵.

注: Toeplitz 矩阵的逆是反向对称矩阵, 但不一定是 Toeplitz 矩阵.

Yule-Walker 方程组

假定 T_n 对称正定, 考虑线性方程组

$$T_n x = -r_n, \tag{2.11}$$

其中 $r_n = [t_1, t_2, \ldots, t_{n-1}, t_n]^T$. 这类线性方程组称为 Yule-Walker 方程组, 其中 t_n 为任意给定的实数.

由于 T_n 对称正定, 所以 $t_0 > 0$. 因此我们可以对 T_n 的对角线元素进行单位化. 不失一般性, 我们假定 T_n 的对角线元素为 1, 即

$$T_n = \begin{bmatrix} 1 & t_1 & \cdots & t_{n-1} \\ t_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_1 \\ t_{n-1} & \cdots & t_1 & 1 \end{bmatrix}.$$

由于方程组右端项的特殊性, 我们可以通过递推来求解.

设 x_k 是 $T_k x = -r_k$ 的解, 下面导出 $T_{k+1} x = -r_{k+1}$ 的解 x_{k+1}. 记

$$x_{k+1} = \begin{bmatrix} z_k \\ \alpha_k \end{bmatrix},$$

则 $T_{k+1} x_{k+1} = -r_{k+1}$ 可写为

$$\begin{bmatrix} T_k & J_k r_k \\ r_k^T J_k & 1 \end{bmatrix} \begin{bmatrix} z_k \\ \alpha_k \end{bmatrix} = - \begin{bmatrix} r_k \\ t_{k+1} \end{bmatrix}.$$

因此可得

$$z_k = T_k^{-1}(-r_k - \alpha_k J_k r_k) = x_k - \alpha_k T_k^{-1} J_k r_k, \tag{2.12}$$

$$\alpha_k = -t_{k+1} - r_k^T J_k z_k. \tag{2.13}$$

由于 T_k 是反向对称矩阵, 故 $T_k^{-1} J_k = J_k T_k^{-1}$. 所以可得

$$z_k = x_k - \alpha_k T_k^{-1} J_k r_k = x_k - \alpha_k J_k T_k^{-1} r_k = x_k + \alpha_k J_k x_k.$$

代入 (2.13) 可得

$$(1 + r_k^T x_k)\alpha_k = -t_{k+1} - r_k^T J_k x_k.$$

又

$$\begin{bmatrix} I & J_k x_k \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} T_k & J_k r_k \\ r_k^T J_k & 1 \end{bmatrix} \begin{bmatrix} I & J_k x_k \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} T_k & 0 \\ 0 & 1 + r_k^T x_k \end{bmatrix},$$

由 T_{k+1} 的对称正定性可知 $1 + r_k^T x_k > 0$, 故可得 x_{k+1} 的计算公式

$$\alpha_k = \frac{-t_{k+1} - r_k^T J_k x_k}{1 + r_k^T x_k}, \qquad z_k = x_k + \alpha_k J_k x_k. \quad k = 1, 2, \ldots. \tag{2.14}$$

运算量为 $\mathcal{O}(k)$. 因此, 我们就可以从一阶 Yule-Walker 方程出发, 利用递推公式 (2.14) 计算 $T_n x = -r_n$ 的解. 总的运算量大约为 $3n^2$.

为了减少运算量, 我们引入一个变量 $\beta_k \triangleq 1 + r_k^T x_k$, 则

$$\begin{aligned} \beta_{k+1} &= 1 + r_{k+1}^T x_{k+1} \\ &= 1 + [r_k^T, t_{k+1}] \begin{bmatrix} x_k + \alpha_k J_k x_k \\ \alpha_k \end{bmatrix} \\ &= 1 + r_k^T x_k + \alpha_k (t_{k+1} + r_k^T J_k x_k) \\ &= (1 - \alpha_k^2)\beta_k. \end{aligned}$$

于是可得求解 Yule-Walker 方程组的 Durbin 算法, 总运算量大约为 $2n^2$.

算法 2.11 求解 Yule-Walker 方程组的 Durbin 算法

1: 输入数据: $t = [t_1, t_2, \ldots, t_n]$ % 注: 这里假定 $t_0 = 1$
2: $x(1) = -t_1, \beta = 1, \alpha = -t_1$
3: **for** for $k = 1$ to $n - 1$ **do**
4: $\quad \beta = (1 - \alpha^2)\beta$
5: $\quad \alpha = -\left(t_{k+1} - \sum_{i=1}^k t_{k+1-i} x(i)\right) / \beta$
6: $\quad x(1:k) = x(1:k) + \alpha x(k:-1:1)$
7: $\quad x(k+1) = \alpha$
8: **end for**

一般右端项的 Toeplitz 线性方程组

考虑一般右端项的方程组

$$T_n x = b,$$

其中 T_n 是对称正定 Toeplitz 矩阵, $b = [b_1, b_2, \ldots, b_n]^T$. 与求解 Yule-Walker 方程组类似, 我们利用递推方法来求解.

假定 x_k 和 y_k 分别是方程组

$$T_k x = [b_1, b_2, \ldots, b_k]^T \triangleq b^{(k)}$$

和

$$T_k y = -[t_1, t_2, \ldots, t_k]^T$$

的解. 设 $x_{k+1} = [z_k^T, \mu_k]^T$ 是 $T_{k+1} x = b^{(k+1)}$ 的解, 则可得

$$\begin{bmatrix} T_k & J_k r_k \\ r_k^T J_k & 1 \end{bmatrix} \begin{bmatrix} z_k \\ \mu_k \end{bmatrix} = \begin{bmatrix} b^{(k)} \\ b_{k+1} \end{bmatrix}.$$

通过计算可得

$$\begin{aligned} z_k &= T_k^{-1} b^{(k)} - \mu_k T_k^{-1} J_k r_k = x_k - \mu_k J_k T_k^{-1} r_k = x_k + \mu_k J_k y_k, \\ \mu_k &= \frac{b_{k+1} - r_k^T J_k x_k}{1 + r_k^T y_k}. \end{aligned}$$

所以, 我们可以先计算 $T_k x = b^{(k)}$ 和 $T_k x = -r_k$ 的解, 然后利用上述公式得到 $T_{k+1} x = b^{(k+1)}$ 的解, 这就是 Levinson 算法, 该算法的总运算量大约为 $4n^2$.

算法 2.12 求解对称正定 Toeplitz 线性方程组的 Levinson 算法

1: 输入数据: $t = [t_1, t_2, \ldots, t_{n-1}]$ 和 $b = [b_1, b_2, \ldots, b_n]$ % 这里假定 $t_0 = 1$
2: $y(1) = -t_1, x(1) = b_1, \beta = 1, \alpha = -t_1$
3: **for** $k = 1$ to $n - 1$ **do**
4: $\quad \beta = (1 - \alpha^2)\beta$
5: $\quad \mu = \left(b_{k+1} - \sum_{i=1}^{k} t_{k+1-i} x(i)\right) / \beta$
6: $\quad x(1:k) = x(1:k) + \alpha x(k:-1:1), x(k+1) = \alpha$
7: $\quad$ **if** $k < n - 1$ **then**
8: $\quad\quad \alpha = -\left(t_{k+1} - \sum_{i=1}^{k} t_{k+1-i} y(i)\right) / \beta$
9: $\quad\quad y(1:k) = y(1:k) + \alpha y(k:-1:1)$
10: $\quad\quad y(k+1) = \alpha$
11: $\quad$ **end if**
12: **end for**

2.3 扰动分析

考虑线性方程组

$$Ax = b.$$

设 x^* 是精确解, $\hat{x}$ 是通过数值计算得到的近似解. 假定 $\hat{x}$ 满足线性方程组

$$(A + \delta A)\hat{x} = b + \delta b.$$

下面讨论 $\delta x \triangleq \hat{x} - x^*$ 的大小, 即向后误差分析.

2.3.1 δx 与 $\hat{x}$ 的关系

定理 2.8 设 $\|\cdot\|$ 是任一向量范数（当该范数作用在矩阵上时就是相应的导出范数）, 则 δx 与 $\hat{x}$ 满足下面的关系式

$$\frac{\|\delta x\|}{\|\hat{x}\|} \leq \|A^{-1}\| \cdot \|A\| \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\| \cdot \|\hat{x}\|} \right).$$

当 $\delta b = 0$ 时, 有

$$\frac{\|\delta x\|}{\|\hat{x}\|} \leq \kappa(A) \frac{\|\delta A\|}{\|A\|} \tag{2.15}$$

证明. 由等式 $(A + \delta A)\hat{x} = b + \delta b = Ax^* - \delta b$ 可知 $A(\hat{x} - x^*) = -\delta A\hat{x} + \delta b$, 即

$$\delta x = A^{-1}(-\delta A\hat{x} + \delta b).$$

所以

$$\|\delta x\| \leq \|A^{-1}\| \cdot (\|\delta A\| \cdot \|\hat{x}\| + \|\delta b\|), \tag{2.16}$$

即

$$\frac{\|\delta x\|}{\|\hat{x}\|} \leq \|A^{-1}\| \cdot \|A\| \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\| \cdot \|\hat{x}\|} \right).$$

若 $\delta b = 0$, 则可得

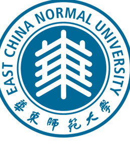
$$\boxed{\frac{\|\delta x\|}{\|\hat{x}\|} \leq \kappa(A) \frac{\|\delta A\|}{\|A\|}}$$

□

2.3.2 δx 与 x^* 的关系

引理 2.3 设 $\|\cdot\|$ 是任一相容范数, $X \in \mathbb{R}^{n\times n}$. 若 $\|X\| < 1$, 则 $I - X$ 可逆, 且有

$$(I - X)^{-1} = \sum_{k=0}^{\infty} X^k \qquad \text{和} \qquad \|(I - X)^{-1}\| \leq \frac{1}{1 - \|X\|}.$$



证明. 先证明级数 $\sum\limits_{k=0}^{\infty} X^k$ 收敛, 即其每个分量都收敛. 记 $x_{ij}^{(k)}$ 为 X^k 的 (i,j) 元素. 由范数的等价性可知, 存在常数 c 使得对任意矩阵 $Y \in \mathbb{R}^{n\times n}$ 都有 $\|Y\|_F \leq c\|Y\|$. 所以

$$\left|x_{ij}^{(k)}\right| \leq \left\|X^k\right\|_F \leq c\left\|X^k\right\| \leq c\left\|X\right\|^k.$$

注意, 这里的常数 c 与 X 和 k 都无关. 由条件 $\|X\| < 1$ 可知, 级数 $\sum\limits_{k=0}^{\infty} c\|X\|^k$ 收敛, 所以级数 $\sum\limits_{k=0}^{\infty} x_{ij}^{(k)}$ 也收敛, 即 $\sum\limits_{k=0}^{\infty} X^k$ 收敛.

因为 $\lim\limits_{k\to\infty} \|X^k\| = 0$, 且 $(I-X)(I+X+X^2+\cdots+X^k) = I - X^{k+1}$, 所以

$$(I-X)\sum_{k=0}^{\infty} X^k = \lim_{k\to\infty}(I - X^{k+1}) = I,$$

即

$$(I-X)^{-1} = \sum_{k=0}^{\infty} X^k,$$

且

$$\|(I-X)^{-1}\| = \|\sum_{k=0}^{\infty} X^k\| \leq \sum_{k=0}^{\infty}\|X^k\| \leq \sum_{k=0}^{\infty}\|X\|^k = \frac{1}{1-\|X\|}.$$

□

由 $(A+\delta A)\hat{x} = b + \delta b$ 可得

$$\begin{aligned}
\delta x &= (A+\delta A)^{-1}(b+\delta b - Ax^* - \delta A x^*) \\
&= (I + A^{-1}\delta A)^{-1}A^{-1}(-\delta A x^* + \delta b).
\end{aligned}$$

假定 $\|\delta A\|$ 很小, 满足 $\|A^{-1}\delta A\| \leq \|A^{-1}\|\cdot\|\delta A\| < 1$, 则由引理 2.3 可得

$$\begin{aligned}
\frac{\|\delta x\|}{\|x^*\|} &\leq \|(I + A^{-1}\delta A)^{-1}\|\cdot\|A^{-1}\|\cdot\left(\|\delta A\| + \frac{\|\delta b\|}{\|x^*\|}\right) \\
&\leq \frac{\|A^{-1}\|}{1-\|A^{-1}\|\cdot\|\delta A\|}\cdot\left(\|\delta A\| + \frac{\|\delta b\|}{\|x^*\|}\right) \\
&= \frac{\|A^{-1}\|\cdot\|A\|}{1-\|A^{-1}\|\cdot\|A\|\cdot\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\|\cdot\|x^*\|}\right) \\
&\leq \frac{\kappa(A)}{1-\kappa(A)\cdot\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right)
\end{aligned}$$

当 $\|\delta A\| \to 0$ 时, 我们可得

$$\boxed{\frac{\|\delta x\|}{\|x^*\|} \leq \frac{\kappa(A)}{1-\kappa(A)\cdot\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right) \to \kappa(A)\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right)} \tag{2.17}$$

定理 2.9 设 $A \in \mathbb{R}^{n\times n}$ 非奇异且 $\|A^{-1}\| \cdot \|\delta A\| < 1$, 则

$$\frac{\|\delta x\|}{\|x^*\|} \le \frac{\kappa(A)}{1-\kappa(A)\cdot\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\delta b\|}{\|b\|}\right).$$

如果 $\|\delta A\| = 0$, 则

$$\frac{1}{\kappa(A)}\frac{\|\delta b\|}{\|b\|} \le \frac{\|\delta x\|}{\|x^*\|} \le \kappa(A)\frac{\|\delta b\|}{\|b\|}.$$

证明. 只需证明左边一个不等式即可. 由于 $\delta A = 0$, 所以 $A\delta x = \delta b$. 两边取范数, 然后同除 $\|x^*\|$ 可得

$$\frac{\|A\|\cdot\|\delta x\|}{\|x^*\|} \ge \frac{\|A\delta x\|}{\|A^{-1}b\|} \ge \frac{\|\delta b\|}{\|A^{-1}\|\cdot\|b\|}.$$

所以结论成立. □

定理 2.10 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, 则有

$$\min\left\{\frac{\|\delta A\|_2}{\|A\|_2} \;:\; A+\delta A \text{ 奇异}\right\} = \frac{1}{\kappa_2(A)}$$

证明. 记 $d \triangleq \min\left\{\|\delta A\|_2 \;:\; A+\delta A \text{ 奇异}\right\}$, 只需证明 $d = \frac{1}{\|A^{-1}\|_2}$.

先证明 $d \ge \frac{1}{\|A^{-1}\|_2}$. 若 $\|\delta A\|_2 < \|A^{-1}\|_2^{-1}$, 则

$$\|A^{-1}\delta A\|_2 \le \|A^{-1}\|_2 \cdot \|\delta A\|_2 < 1.$$

由引理 2.3 可知 $I + A^{-1}\delta A$ 非奇异. 因此 $A+\delta A = A(I+A^{-1}\delta A)$ 也非奇异, 这表明使得 $A+\delta A$ 奇异的 δA 必须满足 $\|\delta A\|_2 \ge \|A^{-1}\|_2^{-1}$, 即

$$d \ge \frac{1}{\|A^{-1}\|_2}.$$

下面证明 $d \le \frac{1}{\|A^{-1}\|_2}$, 即证明存在 δA 满足 $\|\delta A\|_2 = \|A^{-1}\|_2^{-1}$ 使得 $A+\delta A$ 奇异. 由范数的定义可知

$$\|A^{-1}\|_2 = \max_{\|x\|_2=1}\|A^{-1}x\|_2,$$

故存在 x 满足 $\|x\|_2 = 1$ 使得

$$\|A^{-1}\|_2 = \|A^{-1}x\|_2.$$

令 $y = A^{-1}x/\|A^{-1}x\|_2$, 则 $\|y\|_2 = 1$, 且

$$\|xy^T\|_2 = \max_{\|z\|_2=1}\|xy^Tz\|_2 = \max_{\|z\|_2=1}|y^Tz|\cdot\|x\|_2 = \max_{\|z\|_2=1}|y^Tz|.$$

由于 $|y^Tz| \le \|y\|_2\cdot\|z\|_2 = 1$, 且当 $z = y$ 时有 $|y^Tz| = 1$, 所以 $\|xy^T\|_2 = 1$. 构造

$$\delta A = -\frac{xy^T}{\|A^{-1}\|_2},$$

则

$$\|\delta A\|_2 = \frac{\|xy^T\|_2}{\|A^{-1}\|_2} = \frac{1}{\|A^{-1}\|_2}.$$

下面证明 $A + \delta A$ 奇异. 我们只需证明以 $A + \delta A$ 为系数矩阵的齐次线性方程组有非零解. 由于 $\|A^{-1}x\|_2 = \|A^{-1}\|_2$, 容易验证

$$(A + \delta A)y = A\frac{A^{-1}x}{\|A^{-1}x\|_2} - \frac{xy^T}{\|A^{-1}\|_2}y = \frac{x}{\|A^{-1}\|_2} - \frac{x}{\|A^{-1}\|_2} = 0,$$

即 $A + \delta A$ 奇异, 所以 $d \leq \frac{1}{\|A^{-1}\|_2}$.

综上可得

$$d = \min\left\{\|\delta A\|_2 \ : \ A + \delta A \text{ 奇异}\right\} = \frac{1}{\|A^{-1}\|_2}.$$

□

☞ 定理 2.10 的结论对所有 p-范数都成立, 参见 [13, 24].

☞ 度量

$$\mathrm{dist}_p(A) \triangleq \min\left\{\frac{\|\delta A\|_p}{\|A\|_p} \ : \ A + \delta A \text{ 奇异}\right\} = \frac{1}{\kappa_p(A)},$$

表示 A 距离奇异矩阵集合的相对距离.

2.3.3 δx 与残量的关系

这是研究线性方程组的扰动理论的一个较实用的方法.

记残量 (残差) 为 $r = b - A\hat{x}$, 则有

$$\delta x = \hat{x} - x^* = \hat{x} - A^{-1}b = A^{-1}(A\hat{x} - b) = -A^{-1}r,$$

所以可得

$$\boxed{\|\delta x\| \leq \|A^{-1}\| \cdot \|r\|}$$

这个估计式的优点是不用去估计 δA 和 δb 的大小. 由于在实际计算中, r 通常是可以计算的, 因此该估计式比较实用.

定理 2.11 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, $\|\cdot\|$ 为任一算子范数. 记 $r = b - A\hat{x}$, 则

(1) 若存在 $\hat{A}$ 满足 $\hat{A}\hat{x} = b$, 则 $\|\hat{A} - A\| \geq \dfrac{\|r\|}{\|\hat{x}\|}$;

(2) 存在 δA 满足 $\|\delta A\| = \dfrac{\|r\|}{\|\hat{x}\|}$, 使得 $(A + \delta A)\hat{x} = b$.

证明. (1) 由 $\hat{A}\hat{x} = b$ 可知

$$(\hat{A} - A)\hat{x} = b - A\hat{x} = r.$$

所以有

$$\|r\| = \|(\hat{A} - A)\hat{x}\| \leq \|\hat{A} - A\| \cdot \|\hat{x}\|,$$

即

$$\|\hat{A} - A\| \geq \frac{\|r\|}{\|\hat{x}\|}.$$

(2) 以 2-范数为例, 取 $\delta A = \dfrac{r\hat{x}^T}{\|\hat{x}\|_2^2}$ 即可. □

2.3.4 相对扰动分析

前面给出了解的误差 δx 的界是与条件数 $\kappa(A)$ 与 δA 和 δb 成比例的. 在许多情况下, 这个界是令人满意的. 但有时会相差很大, 这个界就不能很好的反映实际计算中解的误差.

例 2.4 设 $A = \begin{bmatrix} \gamma & 0 \\ 0 & 1 \end{bmatrix}$, $b = \begin{bmatrix} \gamma \\ 1 \end{bmatrix}$, 其中 $\gamma > 1$. 则 $Ax = b$ 的精确解为 $x^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, 任何合理的直接法求得的解的误差都很小. 但系数矩阵的谱条件数为 $\kappa_2(A) = \gamma$, 当 γ 很大时, $\kappa_2(A)$ 也很大, 因此误差界 (2.15) 和 (2.17) 可以是很大.

针对这个问题, 我们按分量进行分析. 记

$$\delta A = \begin{bmatrix} \delta a_{11} & \\ & \delta a_{22} \end{bmatrix}, \quad \delta b = \begin{bmatrix} \delta b_1 \\ \delta b_2 \end{bmatrix},$$

并设 $|\delta a_{ij}| \leq \varepsilon |a_{ij}|$, $|\delta b_i| \leq \varepsilon |b_i|$. 则

$$\delta x = \begin{bmatrix} \hat{x}_1 - x_1 \\ \hat{x}_2 - x_2 \end{bmatrix} = \begin{bmatrix} \dfrac{\delta b_1 + b_1}{\delta a_{11} + a_{11}} - 1 \\ \dfrac{\delta b_2 + b_2}{\delta a_{22} + a_{22}} - 1 \end{bmatrix} = \begin{bmatrix} \dfrac{\delta b_1 + \gamma}{\delta a_{11} + \gamma} - 1 \\ \dfrac{\delta b_2 + 1}{\delta a_{22} + 1} - 1 \end{bmatrix} = \begin{bmatrix} \dfrac{\delta b_1 - \delta a_{11}}{\delta a_{11} + \gamma} \\ \dfrac{\delta b_2 - \delta a_{22}}{\delta a_{22} + 1} \end{bmatrix}.$$

故

$$\|\delta x\|_\infty \leq \frac{2\varepsilon}{1-\varepsilon}.$$

如果 $\delta b = 0$, 则

$$\|\delta x\|_\infty \leq \frac{\varepsilon}{1-\varepsilon}.$$

这个界与 (2.15) 或 (2.17) 相差约 γ 倍.

相对条件数

为了得到更好误差界, 我们引入**相对条件数** $\kappa_{cr}(A)$, 即

$$\kappa_{cr}(A) \triangleq \|\, |A^{-1}| \cdot |A| \,\|,$$

有时也称为 Bauer 条件数或 Skeel 条件数.

假定 δA 和 δb 满足 $|\delta A| \leq \varepsilon |A|$ 和 $|\delta b| \leq \varepsilon |b|$. 则由 $(A + \delta A)\hat{x} = b + \delta b$ 可得

$$\begin{aligned} |\delta x| &= |A^{-1}(-\delta A \hat{x} + \delta b)| \\ &\leq |A^{-1}| \cdot (|\delta A| \cdot |\hat{x}| + |\delta b|) \\ &\leq |A^{-1}| \cdot (\varepsilon |A| \cdot |\hat{x}| + \varepsilon |b|) \\ &= \varepsilon |A^{-1}| \cdot (|A| \cdot |\hat{x}| + |b|). \end{aligned} \tag{2.18}$$

若 $\delta b = 0$, 则有

$$\|\delta x\| = \|\, |\delta x| \,\| \leq \varepsilon \|\, |A^{-1}| \cdot |A| \cdot |\hat{x}| \,\| \leq \varepsilon \|\, |A^{-1}| \cdot |A| \,\| \cdot \|\hat{x}\|,$$

即

$$\boxed{\frac{\|\delta x\|}{\|\hat{x}\|} \leq \|\, |A^{-1}| \cdot |A| \,\| \cdot \varepsilon = \kappa_{cr}(A) \cdot \varepsilon} \tag{2.19}$$

相对条件数有下面的性质

引理 2.4 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, $D \in \mathbb{R}^{n\times n}$ 为非奇异对角矩阵, 则

$$\kappa_{cr}(DA) = \kappa_{cr}(A).$$

定理 2.12 设 $A \in \mathbb{R}^{n\times n}$ 非奇异. 使得 $|\delta A| \leq \varepsilon|A|$, $|\delta b| \leq \varepsilon|b|$ 成立, 且满足

$$(A + \delta A)\hat{x} = b + \delta b$$

的最小的 $\varepsilon > 0$ 称为按分量的相对向后误差, 其表达式为

$$\varepsilon = \max_{1\leq i\leq n} \frac{|r_i|}{(|A| \cdot |\hat{x}| + |b|)_i},$$

其中 $r = b - A\hat{x}$.

更多关于数值计算的稳定性和矩阵扰动分析方面的知识, 可以参考 [21, 38, 50].

2.4 误差分析

2.4.1 LU 分解的舍入误差分析

关于 LU 分解的舍入误差分析, 我们有下面的结果.

定理 2.13 假定 $A \in \mathbb{R}^{n\times n}$ 的所有顺序主子式都不为 0, 则带舍入误差的 LU 分解可表示为

$$A = LU + E,$$

其中误差 E 满足

$$|E| \le \gamma_n |L| \cdot |U|.$$

这里 $\gamma_n = \dfrac{n\varepsilon_m}{1 - n\varepsilon_m}$, ε_m 表示机器精度.

证明. 见 [21, page 164]. □

2.4.2 Gauss 消去法的舍入误差分析

引理 2.5 [21] 设 $\hat{y}$ 和 $\hat{x}$ 分别是由向前回代算法 2.5 和向后回代算法 2.6 计算得到的数值解, 则

$$\begin{aligned} &(L + \delta L)\hat{y} = b, \quad |\delta L| \le \gamma_n |L| \\ &(U + \delta U)\hat{x} = \hat{y}, \quad |\delta U| \le \gamma_n |U|. \end{aligned}$$

该引理表明, $\hat{y}$ 和 $\hat{x}$ 只有很小的误差, 因此向前回代算法 2.5和向后回代算法 2.6 都是稳定的. 于是

$$\begin{aligned} b &= (L + \delta L)\hat{y} \\ &= (L + \delta L)(U + \delta U)\hat{x} \\ &= (LU + L\cdot\delta U + \delta L\cdot U + \delta L\cdot\delta U)\hat{x} \\ &= (A - E + L\cdot\delta U + \delta L\cdot U + \delta L\cdot\delta U)\hat{x}. \end{aligned}$$

记 $\delta A = -E + L\cdot\delta U + \delta L\cdot U + \delta L\cdot\delta U$, 则 $\hat{x}$ 是扰动方程 $(A + \delta A)x = b$ 精确解, 且

$$\begin{aligned} |\delta A| &= |-E + L\cdot\delta U + \delta L\cdot U + \delta L\cdot\delta U| \\ &\le |E| + |L|\cdot|\delta U| + |\delta L|\cdot|U| + |\delta L|\cdot|\delta U| \\ &\le (3\gamma_n + \gamma_n^2)|L|\cdot|U| \\ &\le \gamma_{3n}|L|\cdot|U|, \end{aligned}$$

其中 $\gamma_{3n} = \dfrac{3n\varepsilon_m}{1 - 3n\varepsilon_m}$. 两边取范数后可得

$$\|\delta A\| \le 3n\varepsilon_m \|L\|\cdot\|U\|$$

对 1-范数, ∞-范数和 F-范数成立 (2-范数不成立).

根据算法向后稳定性的定义, 要说明带选主元 Gauss 消去法是向后稳定的, 必须要求 $\|\delta A\|$ 是 "小" 的, 即

$$\|\delta A\| = O(\varepsilon_m)\|A\|.$$

数值试验表明, 部分选主元 Gauss 消去法几乎总是保持

$$\|L\| \cdot \|U\| \approx \|A\|.$$

记

$$\rho_n \triangleq \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}$$

为部分选主元 Gauss 消去法的**增长因子**, 其中 $a_{ij}^{(k)}$ 是部分选主元 Gauss 消去法过程第 k 步时 a_{ij} 的值. 由于 $\|L\|_\infty \leq n$, $\|U\|_\infty \leq n\rho_n\|A\|_\infty$, 因此

定理 2.14 设 $\hat{x}$ 是由部分选主元 *Gauss* 法得到的数值解, 则 $\hat{x}$ 满足

$$(A + \delta A)\hat{x} = b, \quad \|\delta A\|_\infty \leq n^2\gamma_{3n}\rho_n\|A\|_\infty. \tag{2.20}$$

所以若 ρ_n 比较小或随着 n 变大时增长比较缓慢, 则当 n 不是很大时, 部分选主元 Gauss 消去法是向后稳定的. 遗憾的是, 理论上无法保证 ρ_n 比较小.

性质 2.1 *[21, page 166]* 部分选主元 *Gauss* 消去法能保证 $\rho_n \leq 2^{n-1}$, 且这个界是可以达到的.

☞ 事实上, (2.20) 中的界几乎总是远远大于真正的 $\|\delta A\|$.

☞ 在绝大多数情况下, 部分选主元 Gauss 消去法是向后稳定的, 但也存在失败的例子.

☞ 全主元 Gauss 消去法比部分选主元 Gauss 消去法更稳定.

2.4.3 部分选主元 Gauss 消去法的舍入误差分析

To be continued ... 参见李大明 [49].

2.5 解的改进和条件数估计

当矩阵 A 是病态时, 即使残量 $r = b - A\hat{x}$ 很小, 所求得的数值解 $\hat{x}$ 仍可能带有较大的误差. 此时需要通过一些方法来提高解的精度.

2.5.1 高精度运算

在计算中, 尽可能采用高精度的运算. 比如, 原始数据是单精度的, 但在计算时都采用双精度运算, 或者更高精度的运算. 但更高精度的运算会带来更大的开销.

2.5.2 矩阵元素缩放 (Scaling)

如果 A 的元素在数量级上相差很大, 则在计算过程中很可能会出现大数与小数的加减运算, 这样就可能会引入更多的舍入误差. 为了避免由于这种情况而导致的舍入误差, 我们可以在求解之前先对矩阵元素进行缩放 (Scaling), 即在矩阵两边同时乘以两个适当的对角矩阵.

例 2.5 考虑线性方程组

$$\begin{bmatrix} -4000 & 2000 & 2000 \\ 2000 & 0.78125 & 0 \\ 2000 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 400 \\ 1.3816 \\ 1.9273 \end{bmatrix}.$$

用部分选主元 Gauss 消去法求解, 计算过程中保留 8 位有效数字, 最后求得的数值解为

$$\tilde{x} = [0.00096365, -0.698496, 0.90042329]^T.$$

而精确解为 $x = [1.9273..., -0.698496..., 0.9004233...]^T$. 数值解的第一个分量存在很大的误差.

我们考虑对矩阵元素进行缩放, 即在方程两边同时乘以一个对角矩阵 $D = \mathrm{diag}(0.00005, 1, 1)$, 然后求解一个新的方程组

$$DADy = Db.$$

最后令 $\tilde{x} = Dy$, 即可求得比较精确的数值解.

设 β 是计算机浮点数的基 (一般为 2), 构造对角矩阵

$$D_1 = \mathrm{diag}(\beta^{r_1}, \beta^{r_2}, \ldots, \beta^{r_n}), \quad D_2 = \mathrm{diag}(\beta^{c_1}, \beta^{c_2}, \ldots, \beta^{c_n}).$$

将原线性方程组 $Ax = b$ 转化为等价方程组 $D_1^{-1}AD_2y = D_1^{-1}b$, 即缩放后的线性方程组. 在对方程组进行缩放时, 需要 $\mathcal{O}(n^2)$ 运算量, 通常不会产生较大的舍入误差.

☞ 对角矩阵 D_1 是对系数矩阵进行缩放, 而 D_2 是对未知量进行缩放.

通过求解缩放后的线性方程组, 我们可以证明 [16]

$$\frac{\|D_2^{-1}(\tilde{x} - x)\|_\infty}{\|D_2^{-1}x\|_\infty} \approx \varepsilon_m \kappa_\infty(D_1^{-1}AD_2),$$

其中 ε_m 是机器精度.

☞ 为了平衡矩阵元素的大小, 一种好的方案是左乘一个对角矩阵 D^{-1} (即对 A 行缩放), 其中 $D_{ii} = \sum_{j=1}^{n} |a_{ij}|$. 然后再执行部分选主元 LU 分解.

2.5.3 迭代改进法

设近似解 $\hat{x}$, 残量 $r = b - A\hat{x}$. 当 $\hat{x}$ 没达到精度要求时, 可以考虑方程 $Az = r$. 如果 z 该方程的精确解, 则

$$A(\hat{x} + z) = A\hat{x} + Az = (b - r) + r = b,$$

因此 $\hat{x} + z$ 就是原方程的精确解. 在实际计算中, 我们只能得到近似解 $\hat{z}$, 但 $\|r - A\hat{z}\|$ 会很小, 特别地, 应该比 $\|r\|$ 更小. 因此 $\hat{x} + \hat{z}$ 应该比 $\hat{x}$ 更接近精确解.

如果新的近似解 $\hat{x} + \hat{z}$ 还不满足精度要求, 则可重复以上过程.

这就是通过迭代来提高解的精度.

算法 2.13 通过迭代改进解的精度

1: 设 $PA = LU$, $\hat{x}$ 是 $Ax = b$ 的近似解
2: **while** 近似解 $\hat{x}$ 不满足精度要求, **do**
3: 　　计算 $r = b - A\hat{x}$
4: 　　求解 $Ly = Pr$, 即 $y = L^{-1}Pr$
5: 　　求解 $Uz = y$, 即 $z = U^{-1}y$
6: 　　令 $\hat{x} = \hat{x} + z$
7: **end while**

由于每次迭代只需计算一次残量和求解两个三角线性方程组, 因此运算量为 $\mathcal{O}(n^2)$. 所以相对来讲还是比较经济的.

☞ 为了提高计算精度, 在计算残量 r 时最好使用原始数据 A, 而不是 $P^T LU$, 因此对 A 做 LU 分解时需要保留矩阵 A, 不能被 L 和 U 覆盖.

☞ 实际计算经验表明, 当 A 病态不是很严重时, 即 $\varepsilon_m \kappa_\infty(A) < 1$, 迭代法可以有效改进解的精度, 最后达到机器精度. 但 $\varepsilon_m \kappa_\infty(A) \geq 1$ 时, 一般没什么效果. 这里 ε_m 表示机器精度.

2.5.4 矩阵条件数估计

To be continued ... 参见李大明 [49].

2.6 课后习题

2.1 设 $A = [a_{ij}] \in \mathbb{R}^{n\times n}$, 且 $a_{11} \neq 0$, 经过一次 Gauss 消去法后得到 $A^{(2)} = \begin{bmatrix} a_{11} & * \\ 0 & A_2 \end{bmatrix}$.
证明: (1) 若 A 对称, 则 A_2 也对称; (2) 若 A 对称正定, 则 A_2 也对称正定.

2.2 设矩阵 $A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 12 \end{bmatrix}$. 计算 A 的 LU 分解, 其中 L 是单位下三角阵, U 是非奇异上三角阵.

2.3 设矩阵 $A = \begin{bmatrix} 4 & 2 & 4 \\ 2 & 37 & 8 \\ 4 & 8 & 14 \end{bmatrix}$. 计算 A 的 Cholesky 分解.

2.4 设矩阵 $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & a \\ 0 & a & 2 \end{bmatrix}$. 问: 当 a 取何值时, A 存在 Cholesky 分解?

2.5 设 $A \in \mathbb{R}^{n\times m}$, 其中 $n \geq m$, 证明: $\|A^TA\|_2 = \|A\|_2^2$.
当 $m = n$ 时, 证明: $\kappa_2(A^TA) = \kappa_2(A)^2$.

2.6 设 $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ 对称正定, 证明: $a_{ij}^2 < a_{ii}a_{jj}$.

2.7 证明 Cholesky 分解 (即定理 2.4) 的唯一性.

2.8 设 $A \in \mathbb{R}^{n\times n}$ 对称非奇异, 且存在分解 $A = LDM^T$, 其中 $L, M \in \mathbb{R}^{n\times n}$ 是单位下三角矩阵, $D \in \mathbb{R}^{n\times n}$ 是对角矩阵. 证明: $L = M$.

2.9 将 $A \in \mathbb{R}^{n\times n}$ 写成分块形式

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

其中 $A_{11} \in \mathbb{R}^{k\times k}$ $(1 \leq k \leq n)$ 非奇异. 我们称矩阵 $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ 为 A 中 A_{11} 的 Schur 补 (通常简称 Schur 补).

(1) 假设 A 存在 LU 分解, 证明: 对于不选主元的 Gauss 消去法, 第 k 步后, A_{22} 已被 S 覆盖.

(2) 假设 $A_{21} = A_{12}^T$, 且 A_{11} 和 $-A_{22}$ 都正定, 证明 A 非奇异.

2.10 设 $A \in \mathbb{R}^{n\times n}$ 严格列对角占优, 即

$$|a_{jj}| > \sum_{i=1,\, i\neq j}^{n} |a_{ij}|.$$

证明:

(1) A 非奇异;

(2) 部分选主元 Gauss 消去法与不选主元的 Gauss 消去法是一样的, 即 Gauss 消去法进行 k 步之后, 剩余的右下角部分仍然是严格列对角占优.

2.11 给定一个非奇异矩阵 A 和一个向量 b.
试证明: 对充分小的 $\|\delta A\|_2$, 存在非零的 δA 和 δb, 使得不等式 (2.16) 中的等号成立.

思考题

2.12 以 1-范数或 ∞-范数为例, 证明定理 2.11 中的结论 (2).

2.13 证明不等式 (2.18) 和 (2.19) 中的等号是可以达到的.

实践题

2.14 根据算法 2.10, 编写求解对角占优三对角线性方程组的追赶法程序.
函数形式: `x=LU_tridiag(a,b,c,f)`
其中 a, b, c 分别是系数矩阵的三条对角线, f 是右端项.

2.15 带状矩阵的 LU 分解. 设 A 是 n 阶带状矩阵, 上带宽为 $L < n$, 下带宽为 $M < n$, 编写一个函数, 计算 A 的 LU 分解 (不带选主元), 并统计运算量.
函数形式: `[L,U]=LU_banded(A,L,M)`

第三章 线性最小二乘问题

广义的最小二乘问题包括线性最小二乘问题, 总体最小二乘问题, 等式约束最小二乘问题, 刚性加权最小二乘问题等等. 它在统计学, 最优化问题, 材料与结构力学, 信号与图像处理等方面都有着广泛的应用, 是计算数学的一个重要研究分支, 也是一个活跃的研究领域.

本讲主要介绍求解线性最小二乘问题的三种常用直接方法: 正规方程法 (也称法方程法), QR 分解法和 SVD 分解法. 一般来说, 正规方程法是最快的, 特别是当 A 的条件数较小时, 正规方程法几乎与其他方法一样精确. 而 SVD 分解法是最慢的, 但结果最可靠.

3.1 引言

考虑**线性最小二乘问题**

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2 \tag{3.1}$$

其中 $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. 问题 (3.1) 的解称为**最小二乘解**.

- 当 $m = n$ 且 A 非奇异时, 这就是一个线性方程组, 解为 $x = A^{-1}b$;
- 当 $m > n$ 时, 约束个数大于未知量个数, 此时我们称问题 (3.1) 为**超定**的 (overdetermined);
- 当 $m < n$ 时, 未知量个数大于约束个数, 此时我们称问题 (3.1) 为**欠定** (或亚定) 的 (underdetermined).

在实际应用中, 我们遇到的多数是超定的, 因此, 这里主要讨论超定线性最小二乘问题的求解.

更多关于最小二乘问题, 可参考 [6, 51].

3.2 正规方程

定理 3.1 设 $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$. 则 $x^* \in \mathbb{R}^n$ 是线性最小二乘问题 (3.1) 的解当且仅当残量 $r = b - Ax^*$ 与 $\mathrm{Ran}(A)$ (值域) 正交, 即 x^* 是下面的**正规方程**的解

$$A^T(b - Ax) = 0 \quad 或 \quad A^T Ax = A^T b. \tag{3.2}$$

证明. 充分性: 设 x^* 是正规方程 (3.2) 的解. 对任意向量 $y \in \mathbb{R}^n$, 由 $(b - Ax^*) \perp \mathrm{Ran}(A)$ 可知

$$\begin{aligned}
\|Ay - b\|_2^2 &= \|(Ax^* - b) + A(y - x^*)\|_2^2 \\
&= \|Ax^* - b\|_2^2 + \|A(y - x^*)\|_2^2 \\
&\geq \|Ax^* - b\|_2^2.
\end{aligned}$$

因此, x^* 是线性最小二乘问题 (3.1) 的解.

必要性: 设 x^* 是线性最小二乘问题 (3.1) 的解. 用反证法, 假定 $z \triangleq A^T(b - Ax^*) \neq 0$. 取 $y = x^* + \alpha z$, 其中 $\alpha > 0$, 则有

$$\|Ay - b\|_2^2 = \|Ax^* - b + \alpha Az\|_2^2 = \|Ax^* - b\|_2^2 - 2\alpha\|z\|_2^2 + \alpha^2\|Az\|_2^2.$$

由于 $\|z\|_2 > 0$, 当 α 充分小时, 有 $2\|z\|_2^2 > \alpha\|Az\|_2^2$, 即上式右端小于 $\|Ax^* - b\|_2^2$. 这与 x^* 是最小二乘解相矛盾. 所以 $z = 0$, 即 $A^T(b - Ax^*) = 0$. □

由定理 3.1 可知, 求线性最小二乘问题 (3.1) 的解等价于求正规方程 (3.2) 的解. 由于

$$A^T b \in \mathrm{Ran}(A^T) = \mathrm{Ran}(A^T A),$$

因此正规方程 $A^T Ax = A^T b$ 是相容 (consistent) 的, 即**最小二乘解总是存在的**. 当 $A^T A$ 非奇异时, 这个解是唯一的.

定理 3.2 设 $A \in \mathbb{R}^{m\times n}$ $(m > n)$. 则 $A^T A$ 对称正定当且仅当 A 是列满秩的, 即 $\mathrm{rank}(A) = n$. 此时, 线性最小二乘问题 (3.1) 的解是唯一的, 其表达式为

$$x = (A^T A)^{-1} A^T b.$$

当 A 列满秩时, 我们就可以使用 Cholesky 分解来求解正规方程, 总的运算量大约为 $mn^2 + \frac{1}{3}n^3 + \mathcal{O}(n^2)$, 其中大部分的运算量 (mn^2) 是用来计算 $A^T A$ (由于 $A^T A$ 对称, 因此只需计算其下三角部分).

最小二乘解的几何含义

根据定理 3.1, 我们可以把 b 写成

$$b = Ax^* + r, \quad 其中 \quad r \perp \mathrm{Ran}(A). \tag{3.3}$$

所以 Ax^* 就是 b 在 $\mathrm{Ran}(A)$ 上的正交投影, 见图 3.1.

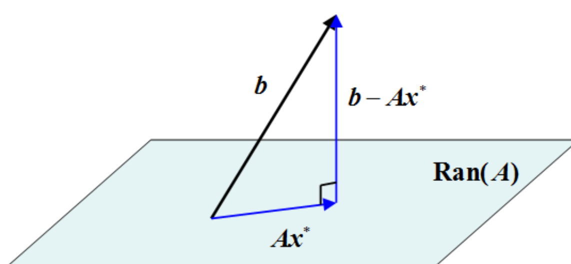


图 3.1 最小二乘解的几何描述

☞ 最小二乘解可能并不唯一, 但分解 (3.3) 总是唯一的.

最小二乘与鞍点问题

由定理 3.1 可知, 线性最小二乘问题 (3.1) 等价于

$$A^T r = 0, \quad r = b - Ax,$$

即

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}. \tag{3.4}$$

这就是线性最小二乘问题 (3.1) 的**增广方程** (augmented system). 事实上, 方程组 (3.4) 是下面方程组的一种特殊情形

$$\begin{bmatrix} B & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix},$$

其中 $B \in \mathbb{R}^{m\times m}$ 对称半正定. 这就是通常所说的**鞍点问题**. 这个方程组存在唯一解当且仅当 A 列满秩且矩阵 $[B, A]$ 行满秩.

当 B 对称正定时, 有 $r = B^{-1}(f - Ax)$, 代入第二个方程可得

$$A^T B^{-1} A x = A^T B^{-1} f - g.$$

这就是**广义的正规方程**. 其所对应的**广义线性最小二乘问题**是

$$\min_{x\in\mathbb{R}^n} \frac{1}{2} \|Ax - f\|_{B^{-1}}^2 + g^T x,$$

其中范数 $\|\cdot\|_{B^{-1}}$ 的定义是 $\|x\|_{B^{-1}}^2 = x^T B^{-1} x$, 这里 B 是对称正定的.

3.3 QR 分解

QR 分解将一个矩阵分解一个正交矩阵 (酉矩阵) 和一个三角矩阵的乘积. QR 分解被广泛应用于线性最小二乘问题的求解和矩阵特征值的计算.

3.3.1 QR 分解的存在唯一性

定理 3.3 (QR 分解) [22] 设 $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$. 则存在一个单位列正交矩阵 $Q \in \mathbb{C}^{m\times n}$ (即 $Q^*Q = I_{n\times n}$) 和一个上三角矩阵 $R \in \mathbb{C}^{n\times n}$, 使得

$$A = QR. \tag{3.5}$$

若 A 列满秩, 则存在一个具有正对角线元素的上三角矩阵 R 使得 (3.5) 成立, 且此时 QR 分解唯一, 即 Q 和 R 都唯一.

证明. 设 $A = [a_1, a_2, \ldots, a_n] \in \mathbb{C}^{m\times n}$. 若 A 列满秩, 即 $\mathrm{rank}(A) = n$. 则 QR 分解 (3.5) 就是对 A 的列向量组进行 Gram-Schmidt 正交化过程的矩阵描述 (见算法 3.1).

算法 3.1 Gram-Schmidt Process

```
1: r_11 = ||a_1||_2
2: q_1 = a_1/r_11
3: for j = 2 to n do
4:     q_j = a_j
5:     for i = 1 to j − 1 do
6:         r_ij = q_i^* a_j     % q_i^* 表示共轭转置
7:         q_j = q_j − r_ij q_i
8:     end for
9:     r_jj = ||q_j||_2
10:    q_j = q_j/r_jj
11: end for
```

如果 A 不是列满秩, 我们可以做类似的正交化过程:

- 如果 $a_1 = 0$, 则令 $q_1 = 0$; 否则令 $q_1 = a_1/\|a_1\|_2$;
- 对于 $j = 2, 3, \ldots, n$, 计算 $\tilde{q}_j = a_j - \sum_{i=1}^{j-1}(q_i^* a_j)q_i$. 如果 $\tilde{q}_j = 0$, 则表明 a_j 可以由 $a_1, a_2, \ldots, a_{j-1}$ 线性表出, 令 $q_j = 0$. 否则令 $q_j = \tilde{q}_j/\|\tilde{q}_j\|_2$.

于是我们有

$$A = QR,$$

其中 $Q = [q_1, q_2, \ldots, q_n]$ 列正交 (但不是单位列正交), 其列向量要么是单位向量, 要么就是零向量. 这里的 $R = [r_{ij}]_{n\times n}$ 是上三角矩阵, 定义如下

$$r_{ij} = \begin{cases} q_i^* a_j, & \text{for } i \leq j \\ 0, & \text{for } i > j \end{cases}$$

如果 Q 的某一列 $q_k = 0$, 则 R 中对应的第 k 行就全为 0.

设 $\text{rank}(A) = l < n$, 则 Q 有 l 个非零列, 设为 $q_{i_1}, q_{i_2}, \dots, q_{i_l}$. 它们形成 $\mathbb{C}^m$ 中的一个单位正交向量组, 所以我们可以将其扩展成 $\mathbb{C}^m$ 中的一组标准正交基, 即

$$q_{i_1}, q_{i_2}, \dots, q_{i_l}, \tilde{q}_1, \dots, \tilde{q}_{m-l}.$$

然后我们用 $\tilde{q}_1$ 替换 Q 中的第一个零列, 用 $\tilde{q}_2$ 替换 Q 中的第二个零列, 依此类推, 将 Q 中的所有零列都替换掉. 将最后得到的矩阵记为 $\tilde{Q}$, 则 $\tilde{Q} \in \mathbb{C}^{m\times n}$ 单位列正交, 且

$$\tilde{Q}R = QR.$$

这是由于 $\tilde{Q}$ 中的新的列向量正好与 R 中的零行相对应. 所以我们有 QR 分解

$$A = \tilde{Q}R.$$

下面证明满秩矩阵 QR 分解的存在唯一性.

存在性: 由于 A 列满秩, 由 Gram-Schmidt 正交化过程 (算法 3.1) 可知, 存在上三角矩阵 $R = [r_{ij}]_{n\times n}$ 满足 $r_{jj} > 0$, 使得 $A = QR$, 其中 Q 单位列正交.

唯一性: 假设 A 存在 QR 分解

$$A = Q_1R_1 = Q_2R_2,$$

其中 $Q_1, Q_2 \in \mathbb{C}^{m\times n}$ 单位列正交, $R_1, R_2 \in \mathbb{C}^{n\times n}$ 为具有正对角元素的上三角矩阵. 则有

$$Q_1 = Q_2R_2R_1^{-1}. \tag{3.6}$$

由于 R_1, R_2 均为上三角矩阵, 所以 $R_2R_1^{-1}$ 也是上三角矩阵, 且其对角线元素为 $R_2(i,i)/R_1(i,i)$, $i = 1, 2, \dots, n$. 由 (3.6) 可得

$$1 = \|Q_1\|_2 = \|Q_2R_2R_1^{-1}\|_2 = \|R_2R_1^{-1}\|_2.$$

所以

$$\frac{R_2(i,i)}{R_1(i,i)} \le 1, \quad i = 1, 2, \dots, n.$$

同理可证 $R_1(i,i)/R_2(i,i) \le 1$. 所以

$$R_1(i,i) = R_2(i,i), \quad i = 1, 2, \dots, n.$$

又 $\|Q_1\|_F^2 = \text{tr}(Q_1^*Q_1) = n$, 所以由 (3.6) 可知

$$\|R_2R_1^{-1}\|_F^2 = \|Q_2R_2R_1^{-1}\|_F^2 = \|Q_1\|_F^2 = n.$$

由于 $R_2R_1^{-1}$ 的对角线元素都是 1, 所以 $R_2R_1^{-1}$ 只能是单位矩阵, 即 $R_2 = R_1$. 因此 $Q_2 = AR_2^{-1} = AR_1^{-1} = Q_1$, 即 A 的 QR 分解是唯一的. □

☞ 有时也将 QR 分解定义为: 存在酉矩阵 $Q \in \mathbb{C}^{n\times n}$ 使得

$$A = Q\begin{bmatrix} R_{11} \\ 0 \end{bmatrix},$$

其中 R_{11} 为上三角矩阵.

由 QR 分解的存在性证明过程可知, 当 A 不是满秩矩阵时, 存在一个置换矩阵 P, 使得 AP 的前 l 列是线性无关的, 其中 $l = \text{rank}(A)$. 因此我们可以对 AP 进行 QR 分解, 于是我们可以得到下面的结论.

推论 3.4 设 $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$. 则存在一个置换矩阵 P, 使得

$$AP = Q\begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}_{n\times n},$$

其中 $Q \in \mathbb{C}^{m\times n}$ 满足 $Q^*Q = I_{n\times n}$, $R_{11} \in \mathbb{C}^{l\times l}$ 是非奇异上三角矩阵.

☞ 如果 A 是实矩阵, 则上面证明中的运算都可以在实数下进行, 因此 Q 和 R 都可以是实矩阵.

☞ 如果 A 是非奇异的方阵, 则 QR 分解也可以用来求解线性方程组 $Ax = b$.

☞ 基于 GS 正交化的 QR 分解算法 3.1 的运算量大约为 $2mn^2$. 在后面, 我们会介绍基于 Household 变换的 QR 分解, 在不需要计算 Q 的情况下, 运算量大约为 $2mn^2 - 2n^3/3$; 如果需要计算 Q, 则需另外大约 $2mn^2 - 2n^3/3$ 运算量.

推论 3.5 (满秩分解) 设 $A \in \mathbb{C}^{m\times n}$ 且 $\text{rank}(A) = l \leq \min\{m, n\}$, 则存在满秩矩阵 $F \in \mathbb{C}^{m\times l}$ 和 $G \in \mathbb{C}^{l\times n}$, 使得

$$A = FG.$$

3.3.2 QR 分解与线性最小二乘问题

这里假定 $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ 是满秩的. 设 A 的 QR 分解为 $A = QR$, 我们用三种不同的方法来推导线性最小二乘问题的解.

- 将 Q 的扩充成一个正交矩阵, 记为 $[Q, \hat{Q}] \in \mathbb{R}^{m\times m}$. 于是有

$$\begin{aligned}
\|Ax - b\|_2^2 &= \left\|[Q, \hat{Q}]^T(Ax - b)\right\|_2^2 \\
&= \left\|[Q, \hat{Q}]^T(QRx - b)\right\|_2^2 \\
&= \left\|\begin{bmatrix} Rx - Q^T b \\ -\hat{Q}^T b \end{bmatrix}\right\|_2^2 \\
&= \|Rx - Q^T b\|_2^2 + \left\|\hat{Q}^T b\right\|_2^2 \\
&\geq \left\|\hat{Q}^T b\right\|_2^2,
\end{aligned}$$

等号成立当且仅当 $Rx = Q^T b$. 所以最小二乘解为

$$x^* = R^{-1}Q^T b.$$

- 将 b 写成 $b = (QQ^T + I - QQ^T)b$, 则

$$Ax - b = Ax - (QQ^T + I - QQ^T)b$$

$$= (Ax - QQ^T b) - (I - QQ^T)b.$$

由 $A = QR$ 和 $Q^T Q = I$ 可知 $A^T(I - QQ^T) = A^T - R^T Q^T QQ^T = A^T - R^T Q^T = 0$, 所以

$$\left(Ax - QQ^T b\right)^T (I - QQ^T)b = -b^T QQ^T (I - QQ^T)b = 0,$$

即 $Ax - QQ^T b$ 与 $(I - QQ^T)b$ 正交. 所以

$$\begin{aligned}
\|Ax - b\|_2^2 &= \|Ax - QQ^T b\|_2^2 + \|(I - QQ^T)b\|_2^2 \\
&= \|Rx - Q^T b\|_2^2 + \|(I - QQ^T)b\|_2^2 \\
&\geq \|(I - QQ^T)b\|_2^2,
\end{aligned}$$

等号成立当且仅当 $Rx = Q^T b$. 所以最小二乘解为

$$x^* = R^{-1} Q^T b.$$

☞ 事实上, Q 的列向量组成 $\mathrm{Ran}(A)$ 的一组标准正交基, 因此 QQ^T 是 $\mathrm{Ran}(A)$ 上的正交投影算子, 且

$$QQ^T b = QR(R^{-1} Q^T b) = QRx^* = Ax^*,$$

即 Ax^* 是 b 在 $\mathrm{Ran}(A)$ 上的正交投影.

- 解正规方程. 由定理 3.1 可知, 最小二乘解为

$$\begin{aligned}
x^* &= (A^T A)^{-1} A^T b \\
&= (R^T Q^T QR)^{-1} R^T Q^T b \\
&= (R^T R)^{-1} R^T Q^T b \\
&= R^{-1} Q^T b.
\end{aligned}$$

☞ 用 QR 分解来求最小二乘解的运算量大约为 $2mn^2$ (如果采用 Householder 变换的话, 运算量大约为 $2mn^2 - \frac{2}{3}n^3$). 当 $m \gg n$ 时, 大约为正规方程的两倍. 当 $m = n$ 时, 几乎相同. 通常 QR 算法比较稳定, 特别是当 A 条件数较大 (病态) 时.

3.4 奇异值分解

奇异值分解 (SVD) 分解是矩阵计算中非常有用的工具之一.

设 $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$, 则 $A^*A \in \mathbb{C}^{n\times n}$ 和 $AA^* \in \mathbb{C}^{m\times m}$ 都是 Hermite 半正定矩阵, 且它们具有相同的非零特征值 (都是正实数).

定理 3.6 (SVD) [16] 设 $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$, 则存在酉矩阵 $U \in \mathbb{C}^{m\times m}$ 和 $V \in \mathbb{C}^{n\times n}$ 使得

$$U^*AV = \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} \quad \text{或} \quad A = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^*, \tag{3.7}$$

其中 $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n) \in \mathbb{R}^{n\times n}$, 且 $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$.

证明. 首先假设 $A \neq 0$, 否则只需令 $\Sigma = 0$ 即可, U 和 V 可以是任意酉矩阵.

下面我们对 m 和 n 用归纳法来证明.

当 $n = 1$ 和 $m \geq n$ 时, 我们取 $\Sigma = \|A\|_2, V = 1, U \in \mathbb{C}^{m\times m}$ 是第一列为 $u_1 = A/\|A\|_2$ 的酉矩阵, 于是

$$A = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^*$$

即为 A 的奇异值分解.

假设 $\mathbb{C}^{(m-1)\times(n-1)}$ 中的矩阵都存在奇异值分解, 下面证明 $A \in \mathbb{C}^{m\times n}$ 也存在有奇异值分解. 由于 $\|A\|_2 = \max\limits_{\|x\|_2=1} \|Ax\|_2$, 所以存在向量 $v \in \mathbb{C}^n$ 满足 $\|v\|_2 = 1$ 使得 $\|A\|_2 = \|Av\|_2$. 令

$$u = \frac{1}{\sigma} Av \in \mathbb{C}^m, \quad \text{其中 } \sigma = \|A\|_2,$$

则 $\|u\|_2 = 1$. 我们将 v 和 u 都扩充成酉矩阵, 即存在 $\tilde{U} \in \mathbb{C}^{m\times(m-1)}$ 和 $\tilde{V} \in \mathbb{C}^{n\times(n-1)}$, 使得 $[u, \tilde{U}] \in \mathbb{C}^{m\times m}$ 和 $[v, \tilde{V}] \in \mathbb{C}^{n\times n}$ 都是酉矩阵. 所以

$$\tilde{U}^*Av = \tilde{U}^*(\sigma u) = 0, \quad u^*Av = u^*(\sigma u) = \sigma.$$

于是

$$\tilde{A} = \left[u, \tilde{U}\right]^* A \left[v, \tilde{V}\right] = \begin{bmatrix} u^*Av & u^*A\tilde{V} \\ \tilde{U}^*Av & \tilde{U}^*A\tilde{V} \end{bmatrix} = \begin{bmatrix} \sigma & u^*A\tilde{V} \\ 0 & \tilde{U}^*A\tilde{V} \end{bmatrix}.$$

又

$$\sigma = \|A\|_2 = \left\|\tilde{A}\right\|_2 = \left\|\tilde{A}^T\right\|_2 \geq \left\|\tilde{A}^T e_1\right\|_2 = \left\|\left[\sigma, u^*A\tilde{V}\right]\right\|_2 = \sqrt{\sigma^2 + \|u^*A\tilde{V}\|_2^2},$$

所以 $\|u^*A\tilde{V}\|_2 = 0$, 即 $u^*A\tilde{V} = 0$. 于是

$$\left[u, \tilde{U}\right]^* A \left[v, \tilde{V}\right] = \begin{bmatrix} \sigma & 0 \\ 0 & A_1 \end{bmatrix},$$

其中 $A_1 = \tilde{U}^*A\tilde{V} \in \mathbb{C}^{(m-1)\times(n-1)}$. 由归纳假设可知, A_1 存在奇异值分解, 设为

$$A_1 = U_1 \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V_1^*,$$

其中 $U_1 \in \mathbb{C}^{(m-1)\times(m-1)}$ 和 $V_1 \in \mathbb{C}^{(n-1)\times(n-1)}$ 都是酉矩阵, $\Sigma_1 \in \mathbb{R}^{(n-1)\times(n-1)}$ 是对角矩阵, 且其对角线元素按降序排列. 令

$$U = \left[u, \tilde{U}\right] \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix}, \quad V = \left[v, \tilde{V}\right] \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix},$$

则 $U \in \mathbb{C}^{m\times m}$ 和 $V \in \mathbb{C}^{n\times n}$ 都是酉矩阵, 且

$$\begin{aligned} U^*AV &= \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix}^* \left[u, \tilde{U}\right]^* A \left[v, \tilde{V}\right] \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix}^* \begin{bmatrix} \sigma & 0 \\ 0 & A_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix} \\ &= \begin{bmatrix} \sigma & 0 \\ 0 & U_1^* A_1 V_1 \end{bmatrix} = \begin{bmatrix} \Sigma \\ 0 \end{bmatrix}, \end{aligned} \tag{3.8}$$

其中 $\Sigma = \begin{bmatrix} \sigma & 0 \\ 0 & \Sigma_1 \end{bmatrix}$. 又

$$\sigma^2 = \|A\|_2^2 = \|U^*AV\|_2^2 = \left\| \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} \right\|_2^2 = \rho\left(\begin{bmatrix} \sigma^2 & 0 \\ 0 & \Sigma_1^2 \end{bmatrix} \right),$$

所以 σ 不小于 Σ_1 中的所有对角线元素, 即 Σ 的对角线元素也是按降序排列. 因此, (3.8) 这就是 A 的奇异值分解.

由归纳法可知, 定理的结论成立. □

☞ 该定理也可以通过 Hermite 半正定矩阵的特征值分解来证明.

定义 3.1 分解 (3.7) 称为 A 的**奇异值分解** (SVD), $\sigma_1, \sigma_2, \ldots, \sigma_n$ 称为 A 的**奇异值**, 它们是矩阵 A^*A 的特征值的平方根.

☞ 我们注意到, 奇异值都是非负实数, 因此对角矩阵 Σ 是实矩阵. 在不做特别说明的情况下, 我们总是假定 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.

☞ 如果 $A \in \mathbb{R}^{m\times n}$ 是实矩阵, 则 U, V 也都可以是实矩阵 [22].

由定理 3.6 可知

$$\operatorname{rank}(A) = \operatorname{rank}(\Sigma).$$

若 $\operatorname{rank}(A) = r \leq n$, 则有

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0.$$

此时 A 的 SVD 可写为

$$A = U \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} V^*, \quad \Sigma_1 = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}.$$

特别地, 如果 $\text{rank}(A)=n$, 则奇异值都是正的, 此时对角矩阵 Σ 非奇异.

矩阵 $U=[u_1,u_2,\dots,u_m]$ 和 $V=[v_1,v_2,\dots,v_n]$ 的列向量分别称为 A 的左奇异向量和右奇异向量, 即存在关系式

$$\begin{aligned} &Av_i=\sigma_i u_i, \quad i=1,2,\dots,n, \\ &A^*u_i=\sigma_i v_i, \quad i=1,2,\dots,n, \\ &A^*u_i=0, \quad i=n+1,n+2,\dots,m. \end{aligned}$$

由定理 3.6 可知

$$A=U\begin{bmatrix}\Sigma\\0\end{bmatrix}V^*=\sigma_1u_1v_1^*+\sigma_2u_2v_2^*+\cdots+\sigma_nu_nv_n^*=\sum_{i=1}^n\sigma_iu_iv_i^*.$$

记 $U_n=[u_1,u_2,\dots,u_n]\in\mathbb{C}^{m\times n}$, 则 U_n 是单位列正交矩阵 (即 $U_n^*U_n=I_{n\times n}$), 且

$$A=U_n\Sigma V^*. \tag{3.9}$$

这就是所谓的细 SVD (thin SVD) [16] 或 降阶 SVD (reduced SVD) [39], 有的文献将 (3.9) 称为奇异值分解.

设 $k<n$, 我们称

$$A_k=\sigma_1u_1v_1^*+\sigma_2u_2v_2^*+\cdots+\sigma_ku_kv_k^*=\sum_{i=1}^k\sigma_iu_iv_i^*$$

为 A 的截断 SVD (truncated SVD).

3.4.1 奇异值分解的性质

下面是关于奇异值的一些基本性质:

定理 3.7 设 $A=U[\Sigma^T,0]^TV^*$ 是 $A\in\mathbb{C}^{m\times n}$ $(m\geq n)$ 的奇异值分解, 则下面结论成立:

(1) 若 $m=n$, 且 A 是 Hermite 的. 设 $A=U\Lambda U^*$ 是 A 的特征值分解, 即 $U^*U=I$, $\Lambda=\text{diag}(\lambda_1,\lambda_2,\dots,\lambda_n)$, 其中 $|\lambda_1|\geq|\lambda_2|\geq\cdots\geq|\lambda_n|$, 则 $A=U\Sigma V$ 是 A 的奇异值分解, 其中 $\sigma_i=|\lambda_i|$, $v_i=\text{sign}(\lambda_i)u_i$, 若 $\lambda_i=0$, 则取 $\text{sign}(\lambda_i)=1$;

(2) A^*A 的特征值是 σ_i^2, 对应的特征向量是 v_i, $i=1,2,\dots,n$;

(3) AA^* 的特征值是 σ_i^2 和 $m-n$ 个零, 对应的特征向量是 u_i, $i=1,2,\dots,m$;

(4) 若 $m=n$, 则矩阵 $H=\begin{bmatrix}0&A^*\\A&0\end{bmatrix}$ 的特征值是 $\pm\sigma_i$, 对应的单位特征向量为 $\dfrac{1}{\sqrt{2}}\begin{bmatrix}v_i\\\pm u_i\end{bmatrix}$;

(5) $\|A\|_2=\sigma_1$, $\|A\|_F=\sqrt{\sigma_1^2+\sigma_2^2+\cdots+\sigma_n^2}$;

(6) 若 $m=n$ 且 A 非奇异, 则 $\|A^{-1}\|_2=\sigma_n^{-1}$, $\kappa_2(A)=\sigma_1/\sigma_n$;

(7) 若 $m=n$, 则 $|\det(A)|=\sigma_1\sigma_2\cdots\sigma_n$;

(8) 若 $\text{rank}(A)=r\leq n$, 则

$$\text{Ran}(A)=\text{span}\{u_1,u_2,\dots,u_r\},\quad \text{Ker}(A)=\text{span}\{v_{r+1},v_{r+2},\dots,v_n\};$$

(9) 设 $x\in\mathbb{C}^n$ 且 $\|x\|_2=1$, 则 $\sigma_n\leq\|Ax\|_2\leq\sigma_1$;

(10) $\sigma_i(A^T)=\sigma_i(A^*)=\sigma_i(A)$;

(11) (酉不变性) 设 $X \in \mathbb{C}^{m\times m}$ 和 $Y \in \mathbb{C}^{n\times n}$ 是酉矩阵, 则 $\sigma_i(X^*AY) = \sigma_i(A)$.

证明. 这些性质的证明都比较简单, 请自己验证. □

下面是关于矩阵 A 的一个低秩逼近.

定理 3.8 设 $A = U_n\Sigma V^*$ 是 $A \in \mathbb{C}^{m\times n}$ 的细奇异值分解. 令 $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^*$, 则 A_k 是

$$\min_{B\in\mathbb{C}^{m\times n},\ \mathrm{rank}(B)=k} \|A-B\|_2 \tag{3.10}$$

的一个解, 且

$$\|A-A_k\|_2 = \sigma_{k+1}.$$

此时, 我们称 A_k 是 A 的一个秩 k 逼近.

证明. 设 $B \in \mathbb{C}^{m\times n}$ 且 $\mathrm{rank}(B) = k$, 则

$$\dim(\mathrm{Ker}(B)) + \dim(\mathrm{span}\{V_{k+1}\}) = (n-k) + (k+1) = n+1 > n,$$

其中 $V_{k+1} = [v_1, v_2, \dots, v_{k+1}] \in \mathbb{C}^{n\times k}$. 所以 $\mathrm{Ker}(B)$ 与 $\mathrm{span}\{V_{k+1}\}$ 有非零公共元素. 令 $0 \neq x \in \mathrm{Ker}(B) \cap \mathrm{span}\{V_{k+1}\}$, 不失一般性, 我们假设 $\|x\|_2 = 1$. 故存在 $y \in \mathbb{C}^{k+1}$ 满足 $\|y\|_2 = 1$ 使得 $x = V_{k+1}y$. 于是

$$\begin{aligned}\|A-B\|_2^2 \geq \|(A-B)x\|_2^2 = \|Ax\|_2^2 = \|U_n\Sigma V^* V_{k+1} y\|_2^2 \\ = \left\|\Sigma \begin{bmatrix} I_{k+1} \\ 0 \end{bmatrix} y\right\|_2^2 = \left\|\begin{bmatrix} \Sigma_1 y \\ 0 \end{bmatrix}\right\|_2^2 = \sum_{i=1}^{k+1} \sigma_i^2 y_i^2 \geq \sigma_{k+1}^2,\end{aligned}$$

其中 $\Sigma_1 = \mathrm{diag}(\sigma_1, \sigma_2, \dots, \sigma_{k+1})$. 这里我们利用了性质 $\sum_{i=1}^{k+1} y_i^2 = \|y\|_2^2 = 1$. 所以

$$\min_{B\in\mathbb{C}^{m\times n},\ \mathrm{rank}(B)=k} \|A-B\|_2 \geq \sigma_{k+1}.$$

又 $\mathrm{rank}(A_k) = k$, 且

$$\|A-A_k\|_2 = \left\|\sum_{i=k+1}^{n} \sigma_i u_i v_i^*\right\|_2 = \left\| U_n \begin{bmatrix} 0 & & & & \\ & \ddots & & & \\ & & 0 & & \\ & & & \sigma_{k+1} & & \\ & & & & \ddots & \\ & & & & & \sigma_n \end{bmatrix} V^* \right\|_2 = \sigma_{k+1},$$

所以

$$\min_{B\in\mathbb{C}^{m\times n},\ \mathrm{rank}(B)=k} \|A-B\|_2 = \sigma_{k+1},$$

且 A_k 是问题 (3.10) 的一个解. □

☞ 对于 Frobenius 范数, 我们有类似的结论, 见习题 3.5.

如果 $\mathrm{rank}(A) = k \le n$, 则 $\sigma_i = 0, i = k+1, k+2, \ldots, n$, 所以

$$A = \sum_{i=1}^{k} \sigma_i u_i v_i^* = A_k. \tag{3.11}$$

定理 3.9 (Weyl) [36, page 67] 设 $A, B \in \mathbb{C}^{m\times n}$ $(m \ge n)$, 且 $\mathrm{rank}(B) = k$. 则有

$$\max_{x\in \mathrm{Ker}(B),\ \|x\|_2=1} \|Ax\|_2 \ge \sigma_{k+1}(A), \tag{3.12}$$

和

$$\min_{x\in \mathrm{Ker}(B),\ \|x\|_2=1} \|Ax\|_2 \le \sigma_{n-k}(A). \tag{3.13}$$

因此,

$$\sigma_1(A-B) \ge \sigma_{k+1}(A), \quad \sigma_n(A+B) \le \sigma_{n-k}(A) \tag{3.14}$$

且

$$\sigma_{i+j-1}(A) \le \sigma_i(B) + \sigma_j(A-B), \quad i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, n-i+1. \tag{3.15}$$

证明. 不等式 (3.12) 也可以写为: 设 $\mathcal{L}$ 是 $\mathbb{C}^n$ 中的任意一个 $n-k$ 维子空间, 则有

$$\max_{x\in \mathcal{L},\ \|x\|_2=1} \|Ax\|_2 \ge \sigma_{k+1}(A).$$

该不等式的证明与定理 3.8 的证明类似.

下面证明结论 (3.13). 令 $\tilde{V}_{k+1} = [v_{n-k}, v_{n-k+1}, \ldots, v_n] \in \mathbb{C}^{n\times(k+1)}$. 类似地, 存在向量 $y \in \mathbb{C}^{k+1}$ 满足 $\|y\|_2 = 1$, 使得 $x = \tilde{V}_{k+1} y \in \mathrm{Ker}(B)$. 于是

$$\|Ax\|_2^2 = \|U_n \Sigma V^* V_{k+1} y\|_2^2 = \left\| \Sigma \begin{bmatrix} 0 \\ I_{k+1} \end{bmatrix} y \right\|_2^2 = \left\| \begin{bmatrix} 0 \\ \tilde{\Sigma}_{k+1} \end{bmatrix} y \right\|_2^2 = \sum_{i=1}^{k+1} \sigma_{n-k-1+i}^2 y_i^2 \le \sigma_{n-k}^2,$$

其中 $\tilde{\Sigma}_{k+1} = \mathrm{diag}(\sigma_{n-k}, \sigma_{n-k+1}, \ldots, \sigma_n)$. 所以

$$\min_{x\in \mathrm{Ker}(B),\ \|x\|_2=1} \|Ax\|_2 \le \sigma_{n-k}(A).$$

不等式 (3.14) 可由 (3.12), (3.13) 以及定理 3.7 中的性质 9 得到.

下面证明不等式 (3.15). 首先证明 $i = j = 1$ 时结论成立. 事实上, 我们有

$$\sigma_1(A) = \|A\|_2 = \|B + (A-B)\|_2 \le \|B\|_2 + \|A-B\|_2 = \sigma_1(B) + \sigma_1(A-B).$$

令 $C = A - B$, 并设 B_{i-1} 和 C_{j-1} 分别是 B 和 C 的秩 $i-1$ 和秩 $j-1$ 逼近. 则

$$\sigma_1(B - B_{i-1}) = \|B - B_{i-1}\|_2 = \sigma_i(B), \quad \sigma_1(C - C_{i-1}) = \|C - C_{i-1}\|_2 = \sigma_j(C).$$

所以

$$\sigma_i(B) + \sigma_j(C) = \sigma_1(B - B_{i-1}) + \sigma_1(C - C_{j-1})$$

$$
\begin{aligned}
&= \|B - B_{i-1}\|_2 + \|C - C_{j-1}\|_2 \\
&\geq \|A - B_{i-1} - C_{j-1}\|_2 \\
&= \sigma_1(A - B_{i-1} - C_{j-1}).
\end{aligned}
$$

又 $\operatorname{rank}(B_{i-1} + C_{j-1}) \leq i + j - 2$, 所以由不等式 (3.14) 可知

$$\sigma_i(B) + \sigma_j(C) = \sigma_1(A - B_{i-1} - C_{j-1}) \geq \sigma_{i+j-1}(A).$$

□

根据定理 3.9, 我们可以得到矩阵奇异值的最小最大定理.

定理 3.10 设 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ 是矩阵 $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$ 的奇异值, 则

$$\sigma_k(A) = \min_{\dim(\mathcal{L})=n-k+1} \ \max_{x\in\mathcal{L},\, \|x\|_2=1} \|Ax\|_2, \quad k = 1, 2, \ldots, n$$

和

$$\sigma_k(A) = \max_{\dim(\mathcal{L})=k} \ \min_{x\in\mathcal{L},\, \|x\|_2=1} \|Ax\|_2, \quad k = 1, 2, \ldots, n,$$

其中 $\mathcal{L}$ 表示 $\mathbb{C}^n$ 的一个子空间.

证明. 见习题 3.3. (也可以利用 Hermite 矩阵的特征值的最小最大定理) □

引理 3.1 (交错不等式) [23, page 149] 设 $A \in \mathbb{C}^{m\times n}$, A_r 是由 A 去除 r 列 (或 r 行) 后得到的子矩阵, 则

$$\sigma_i(A) \geq \sigma_i(A_r) \geq \sigma_{i+r}(A), \quad i = 1, 2, \ldots, \min\{m, n\}.$$

这里, 当下标 k 大于矩阵的维数时, 我们令 $\sigma_k = 0$.

更进一步, 如果 $B \in \mathbb{C}^{(m-r)\times(n-s)}$ 是 A 的子矩阵, 则

$$\sigma_i(A) \geq \sigma_i(B) \geq \sigma_{i+r+s}(A).$$

证明. 只需证明 $r = 1$ 的情形, 其余情形可以通过递推实现.

假设 A_1 是由 A 中去除第 k 列后的子矩阵. 令 $e_k \in \mathbb{C}^n$ 表示 k-th 标准单位向量, 即第 k 个元素是 1, 其它均为 0. 由定理 3.10 可知

$$\sigma_k(A) = \min_{\dim(\mathcal{L})=n-k+1} \ \max_{x\in\mathcal{L},\, \|x\|_2=1} \|Ax\|_2 \geq \min_{\dim(\mathcal{L})=n-k+1} \ \max_{x\in\mathcal{L},\, \|x\|_2=1,\, x\perp e_k} \|Ax\|_2$$

若 $x \perp e_k$, 则 $x_k = 0$, 所以

$$Ax = A_1\tilde{x},$$

其中 $\tilde{x} = [x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n]^T \in \mathbb{C}^{n-1}$. 故条件 “$x \in \mathcal{L}$, $\|x\|_2 = 1$, $x \perp e_k$” 就等价于 “$x \in \tilde{\mathcal{L}}$, $\|\tilde{x}\|_2 = 1$”, 其中 $\tilde{\mathcal{L}} \subset \mathbb{C}^{n-1}$ 是由 $\mathcal{L}$ 中所有向量去除第 k 的分量后组成的集合. 因此 $\dim(\tilde{\mathcal{L}}) = \dim(\mathcal{L}) = n-k+1$ 或 $\dim(\tilde{\mathcal{L}}) = \dim(\mathcal{L}) - 1 = n-k$. 又

$$\min_{\dim(\tilde{\mathcal{L}})=n-k+1} \ \max_{\tilde{x}\in\tilde{\mathcal{L}},\, \|\tilde{x}\|_2=1} \|A_1\tilde{x}\|_2 \geq \min_{\dim(\tilde{\mathcal{L}})=n-k} \ \max_{\tilde{x}\in\tilde{\mathcal{L}},\, \|\tilde{x}\|_2=1} \|A_1\tilde{x}\|_2,$$

且 $A_1 \in \mathbb{C}^{m\times(n-1)}$, 所以

$$\begin{aligned}\sigma_k(A) &= \min_{\dim(\mathcal{L})=n-k+1} \max_{x\in\mathcal{L},\ \|x\|_2=1} \|Ax\|_2 \\ &\geq \min_{\dim(\mathcal{L})=n-k+1} \max_{x\in\mathcal{L},\ \|x\|_2=1, x\perp e_k} \|Ax\|_2 \\ &= \min_{\dim(\tilde{\mathcal{L}})=(n-1)-k+1} \max_{\tilde{x}\in\tilde{\mathcal{L}},\ \|\tilde{x}\|_2=1} \|A_1\tilde{x}\|_2 \\ &= \sigma_k(A_1).\end{aligned}$$

同理可得

$$\begin{aligned}\sigma_{k+1}(A) &= \max_{\dim(\mathcal{L})=k+1} \min_{x\in\mathcal{L},\ \|x\|_2=1} \|Ax\|_2 \\ &\leq \max_{\dim(\mathcal{L})=k+1} \min_{x\in\mathcal{L},\ \|x\|_2=1, x\perp e_k} \|Ax\|_2 \\ &= \max_{\dim(\tilde{\mathcal{L}})=k} \min_{\tilde{x}\in\tilde{\mathcal{L}},\ \|\tilde{x}\|_2=1} \|A_1\tilde{x}\|_2 \\ &= \sigma_k(A_1).\end{aligned}$$

我们注意到, 在前面的证明中, 并没有要求 $m \geq n$. 因此对于 $m < n$ 的情形, 上面的结论仍然成立.

如果 A_1 是由 A 中去除第 k 行后的子矩阵. 由于 A^* 与 A 具有相同的奇异值, 因此只需将上面的讨论运用到 A^* 上即可.

□

引理 3.2 [23, page 170] 设 $A \in \mathbb{C}^{n\times n}$, $1 \leq k \leq n$, 则对任意的单位列正交矩阵 $U_k \in \mathbb{C}^{m\times k}$ 和 $V_k \in \mathbb{C}^{n\times k}$, 有

$$\sigma_i(U_k^* A V_k) \leq \sigma_i(A), \quad i = 1, 2, \dots, k. \tag{3.16}$$

因此,

$$|\det(U_k^* A V_k)| \leq \sigma_1(A)\sigma_2(A)\cdots\sigma_k(A). \tag{3.17}$$

证明. 我们将 U_k 和 V_k 都扩充成酉矩阵 $U \in \mathbb{C}^{m\times m}$ 和 $V \in \mathbb{C}^{n\times n}$, 则 $U_k^* A V_k$ 是 $U^* AV$ 的子矩阵. 由引理 3.1 可知,

$$\sigma_i(U_k^* A V_k) \leq \sigma_i(U^* AV).$$

又 $U^* AV$ 与 A 有相同的奇异值, 故结论 (3.16) 成立.

利用定理 3.7 中的性质 7, 即可得到不等式 (3.17). □

定理 3.11 (Weyl 不等式, 1949) [23, page 171] 设 $A \in \mathbb{C}^{n\times n}$, 其奇异值和特征值分别为 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ 和 $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$. 则

$$|\lambda_1\lambda_2\cdots\lambda_k| \leq \sigma_1\sigma_2\cdots\sigma_k, \quad k = 1, 2, \dots, n$$

且当 $k = n$ 时, 等号成立.

证明. 设 $A = URU^*$ 是 A 的 Schur 分解, 其中 $U \in \mathbb{C}^{n\times n}$ 是酉矩阵, $R \in \mathbb{C}^{n\times n}$ 是上三角矩阵, 且其对角线元素依次为 $\lambda_1, \lambda_2, \ldots, \lambda_n$. 取 U 的前 k 列组成一个单位列正交矩阵 U_k, 则 $R_k \triangleq U_k^* A U_k$ 为 $U^*AU = R$ 的 k 阶顺序主子矩阵, 即 R_k 也是上三角矩阵, 且其对角线元素依次为 $\lambda_1, \lambda_2, \ldots, \lambda_k$. 所以由引理 3.2 可得

$$|\lambda_1\lambda_2\cdots\lambda_k| = |\det(R_k)| = |\det(U_k^* A U_k)| \le \sigma_1\sigma_2\cdots\sigma_k.$$

当 $k = n$ 时,

$$|\lambda_1\lambda_2\cdots\lambda_n| = |\det(A)| = \sigma_1\sigma_2\cdots\sigma_n.$$

□

下面的定理是关于奇异值的一个扰动性质.

定理 3.12 [36, page 69] 设 $A, E \in \mathbb{C}^{m\times n}$ $(m \ge n)$. 则

$$|\sigma_i(A+E) - \sigma_i(A)| \le \|E\|_2, \quad i = 1, 2, \ldots, n.$$

证明. 直接利用不等式 (3.15) 即可, 留作习题 3.4. □

下面是定理 3.12 的一个推论, 也是 SVD 的一个应用.

推论 3.13 设 $A \in \mathbb{C}^{n\times n}$, $\|\cdot\|$ 是任意一个相容矩阵范数, 则对任意的 $\varepsilon > 0$, 总存在一个矩阵 A_ε 使得 $\|A - A_\varepsilon\| \le \varepsilon$, 其中 A_ε 具有互不相同的特征值.

由推论 3.13 可知, 可对角化矩阵在所有矩阵组成的集合中是稠密的.

3.4.2 奇异值分解与线性最小二乘问题

设 $A \in \mathbb{R}^{m\times n}$ 列满秩, $A = U\begin{bmatrix}\Sigma \\ 0\end{bmatrix}V^*$ 是 A 的奇异值分解. 令 U_n 为 U 的前 n 列组成的矩阵, 即 $U = [U_n, \tilde{U}]$, 则

$$\begin{aligned}
\|Ax - b\|_2^2 &= \left\| U \begin{bmatrix}\Sigma \\ 0\end{bmatrix} V^T x - b \right\|_2^2 \\
&= \left\| \begin{bmatrix}\Sigma \\ 0\end{bmatrix} V^T x - [U_n, \tilde{U}]^T b \right\|_2^2 \\
&= \left\| \begin{bmatrix}\Sigma V^T x - U_n^T b \\ -\tilde{U}^T b\end{bmatrix} \right\|_2^2 \\
&= \|\Sigma V^T x - U_n^T b\|_2^2 + \|\tilde{U}^T b\|_2^2 \\
&\ge \|\tilde{U}^T b\|_2^2,
\end{aligned}$$

等号当且仅当 $\Sigma V^T x - U_n^T b = 0$ 时成立, 即

$$x = (\Sigma V^T)^{-1} U_n^T b = V\Sigma^{-1} U_n^T b.$$

这就是线性最小二乘问题 (3.1) 的解.

3.5 最小二乘扰动分析

定理 3.14 设 $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ 且 $\operatorname{rank}(A) = n$. 设 x 是线性最小二乘问题 (3.1) 的解, $\tilde{x}$ 极小化 $\|(A+\delta A)\tilde{x} - (b+\delta b)\|_2$, 则

$$\frac{\|\tilde{x}-x\|_2}{\|x\|_2} \leq \varepsilon \cdot \left\{ \frac{2\kappa_2(A)}{\cos\theta} + \kappa_2^2(A)\tan\theta \right\} + O(\varepsilon^2),$$

其中 $\kappa_2(A) = \sigma_1(A)/\sigma_n(A)$, θ 为 b 与 Ax 之间的夹角 (或 b 与 $\operatorname{Ran}(A)$ 的夹角),

$$\varepsilon \triangleq \max\left\{ \frac{\|\delta A\|_2}{\|A\|_2}, \frac{\|\delta b\|_2}{\|b\|_2} \right\},$$

并假定 $\varepsilon \cdot \kappa_2(A) < 1$ (确保 $A + \delta A$ 满秩, 从而 $\tilde{x}$ 唯一确定).

我们记

$$\kappa_{LS} \triangleq \frac{2\kappa_2(A)}{\cos\theta} + \kappa_2^2(A)\tan\theta,$$

这就是最小二乘问题的条件数. 当 $\theta = 0$ 时, $b \in \operatorname{Ran}(A)$, 此时 $\kappa_{LS} = 2\kappa_2(A)$; 当 $\theta = \pi/2$ 时, $b \perp \operatorname{Ran}(A)$, 此时最小二乘解为 $x = 0$, 而 $\kappa_{LS} = \infty$; 当 $0 < \theta < \pi/2$ 时, $\kappa_{LS} = O(\kappa_2^2(A))$.

定义残量 $r = b - Ax$, $\tilde{r} = (b+\delta b) - (A+\delta A)\tilde{x}$, 我们有下面的性质 [21]

$$\frac{\|\tilde{r}-r\|_2}{\|r\|_2} \leq 1 + 2\varepsilon \cdot \kappa_2(A).$$

当我们使用 QR 分解或 SVD 分解求解最小二乘问题时, 由于采用的是正交变换, 它们都是数值稳定的. 而正规方程涉及求解方程组 $A^TAx = A^Tb$, 其精度依赖于条件数 $\kappa_2(A^TA) = \kappa_2^2(A)$, 其误差总是以 $\kappa_2^2(A)$ 倍数增长. 因此当 A 的条件数较大时, 正规方程就会不太精确.

3.6 初等变换矩阵

矩阵计算的一个基本思想就是把较复杂的问题转化为等价的较简单的, 易于求解问题. 而完成这个转化的基本工具就是初等变换矩阵, 其中使用较多的有三种: Gauss 变换, Householder 变换和 Givens 变换.

3.6.1 初等矩阵

我们考虑初等矩阵

$$E(u, v, \tau) = I - \tau uv^*,$$

其中 $u, v \in \mathbb{C}^n$ 是非零向量, τ 是一个非零复数. 事实上, $E(u, v, \tau)$ 是单位矩阵的一个秩 1 扰动.

定理 3.15 设 $E(u, v, \tau)$ 是一个初等矩阵, 我们有

(1) $\det(E(u, v, \tau)) = 1 - \tau v^* u$;

(2) 若 $1 - \tau v^* u \neq 0$, 则 $E(u, v, \tau)$ 非奇异, 且

$$(E(u, v, \tau))^{-1} = E(u, v, \gamma), \quad \text{其中 } \gamma = \frac{\tau}{\tau v^* u - 1}.$$

证明. (1) 易知

$$\begin{bmatrix} I & 0 \\ v^* & 1 \end{bmatrix} \begin{bmatrix} I - \tau uv^* & -\tau u \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ -v^* & 1 \end{bmatrix} = \begin{bmatrix} I & -\tau u \\ 0 & 1 - \tau v^* u \end{bmatrix}.$$

由行列式的乘法可知

$$\det\left(\begin{bmatrix} I & 0 \\ v^* & 1 \end{bmatrix}\right) \cdot \det\left(\begin{bmatrix} I - \tau uv^* & -\tau u \\ 0 & 1 \end{bmatrix}\right) \cdot \det\left(\begin{bmatrix} I & 0 \\ -v^* & 1 \end{bmatrix}\right) = \det\left(\begin{bmatrix} I & -\tau u \\ 0 & 1 - \tau v^* u \end{bmatrix}\right).$$

所以

$$\det\big(E(u, v, \tau)\big) = \det(I - \tau uv^*) = 1 - \tau v^* u.$$

(2) 若 $1 - \tau v^* u \neq 0$, 则 $\det\big(E(u, v, \tau)\big) \neq 0$, 所以 $E(u, v, \tau)$ 非奇异. 通过直接计算可知

$$\begin{aligned} E(u, v, \tau)E(u, v, \gamma) &= I - \tau uv^* - \gamma(1 - \tau v^* u)uv^* \\ &= I - \tau uv^* - \frac{\tau}{\tau v^* u - 1}(1 - \tau v^* u)uv^* \\ &= I. \end{aligned}$$

□

定理 3.16 设 $A \in \mathbb{C}^{n \times n}$, 则 A 非奇异当且仅当 A 可以分解成若干个初等矩阵的乘积.

3.6.2 Gauss 变换

设 $l_j = (0, \ldots, 0, l_{j+1,j}, \ldots, l_{n,j})^T, j = 1, 2, \ldots, n$, 则 Gauss 变换 定义为

$$L(l_j) \triangleq E(l_j, e_j, -1) = I + l_j e_j^T = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & l_{j+1,j} & 1 & & \\ & & \vdots & & \ddots & \\ & & l_{n,j} & & & 1 \end{bmatrix},$$

向量 l_j 称为 Gauss 向量 [16]. 由定理 3.15 可知

$$\det(L(l_j)) = 1, \quad (L(l_j))^{-1} = E(l_j, e_j, 1) = E(-l_j, e_j, -1) = L(-l_j).$$

Gauss 变换主要用于矩阵的 LU 分解.

3.6.3 Householder 变换

定义 3.2 我们称矩阵

$$H = I - \frac{2}{v^* v} v v^* = I - \frac{2}{\|v\|_2^2} v v^*, \quad 0 \neq v \in \mathbb{C}^n, \tag{3.18}$$

为 Householder 矩阵 (或 Householder 变换, 或 Householder 反射), 向量 v 称为 Householder 向量. 我们通常将矩阵 (3.18) 记为 $H(v)$.

从几何上看, 一个 Householder 变换就是一个关于超平面 $\text{span}\{v\}^\perp$ 的反射. 对任意一个向量 $x \in \mathbb{C}^n$, 可将其写为

$$x = \frac{v^* x}{v^* v} v + y \triangleq \alpha v + y,$$

其中 $\alpha v \in \text{span}\{v\}, y \in \text{span}\{v\}^\perp$. 则

$$Hx = x - \frac{2}{v^* v} v v^* x = x - 2\alpha v = -\alpha v + y,$$

即 Hx 与 x 在 $\text{span}\{v\}^\perp$ 方向有着相同的分量, 而在 v 方向的分量正好相差一个符号. 也就是说, Hx 是 x 关于超平面 $\text{span}\{v\}^\perp$ 的镜面反射, 见图 3.2. 因此, Householder 矩阵也称为反射矩阵.

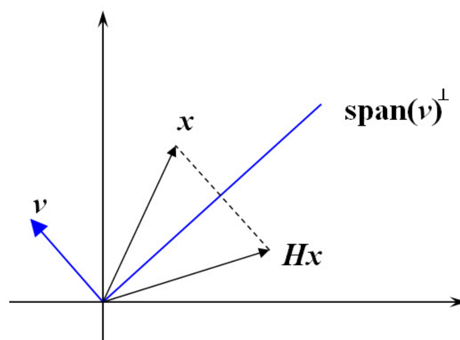


图 3.2 Householder 变换的几何意义

下面是关于 Householder 矩阵的几个基本性质.

定理 3.17 设 $H \in \mathbb{C}^{n\times n}$ 是一个 *Householder* 矩阵, 则

(1) $H^* = H$, 即 H *Hermite* 的;

(2) $H^*H = I$, 即 H 是酉矩阵;

(3) $H^2 = I$, 所以 $H^{-1} = H$;

(4) $\det(H) = -1$;

(5) H 有两个互异的特征值: $\lambda = 1$ 和 $\lambda = -1$, 其中 $\lambda = 1$ 的代数重数为 $n - 1$.

Householder 矩阵的一个非常重要的应用就是可以将一个向量除第一个元素以外的所有元素都化为零. 我们首先给出一个引理.

引理 3.3 设 $x, y \in \mathbb{C}^n$ 为任意两个互异的向量, 则存在一个 *Householder* 矩阵 H 使得 $y = Hx$ 的充要条件是 $\|x\|_2 = \|y\|_2$ 且 $x^*y \in \mathbb{R}$.

证明. 若 $\|x\|_2 = \|y\|_2$ 且 $x^*y \in \mathbb{R}$, 则 $y^*y = x^*x$ 且 $x^*y = y^*x$. 于是

$$\|x - y\|_2^2 = (x - y)^*(x - y) = x^*x - y^*x - x^*y + y^*y = 2(x^*x - y^*x).$$

令 $v = x - y$, 则有

$$H(v)x = x - \frac{2(x - y)(x - y)^*x}{\|x - y\|_2^2} = x - \frac{2(x - y)(x^*x - y^*x)}{2(x^*x - y^*x)} = y,$$

即存在 Householder 矩阵 $H(v)$ 使得 $y = H(v)x$.

反之, 如果存在 Householder 矩阵 H 使得 $y = Hx$, 由于 H 是 Hermite 的, 所以 $x^*y = x^*Hx \in \mathbb{R}$. 又因为 H 是酉矩阵, 所以 $\|y\|_2 = \|Hx\|_2 = \|x\|_2$. □

由引理 3.3, 我们可以立即得到下面的定理.

定理 3.18 设 $x = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$ 是一个非零向量, 则存在 *Householder* 矩阵 H 使得 $Hx = \alpha e_1$, 其中 $\alpha = \|x\|_2$ (或 $\alpha = -\|x\|_2$), $e_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^n$.

设 $x = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$ 是一个实的非零向量, 下面讨论如何计算其对应的 Householder 向量. 根据引理 3.3, H 所对应的 Householder 向量为

$$v = x - \alpha e_1 = [x_1 - \alpha, x_2, \ldots, x_n]^T.$$

在实际计算中, 为了尽可能地减少舍入误差, 我们通常避免两个相近的数做减运算, 否则就会损失有效数字. 因此, 我通常取

$$\alpha = -\operatorname{sign}(x_1) \cdot \|x\|_2.$$

事实上, 我们也可以取 $\alpha = \operatorname{sign}(x_1)\|x\|_2$, 但此时为了减少舍入误差, 我们可以通过下面的公式来计算 v 的第一个分量 v_1

$$\alpha = \operatorname{sign}(x_1)\|x\|_2, \quad v_1 = x_1 - \alpha = \frac{x_1^2 - \|x\|_2^2}{x_1 + \alpha} = \frac{-(x_2^2 + x_3^2 + \cdots + x_n^2)}{x_1 + \alpha}.$$

无论怎样选取 α, 我们都有 $H = I - \beta vv^*$ 其中

$$\beta = \frac{2}{v^* v} = -\frac{1}{\alpha v_1}.$$

在实数域中计算 Householder 向量 v 的算法如下, 总运算量大约为 $3n$.

算法 3.2 计算 Householder 向量

% Given $x \in \mathbb{R}^n$, compute $v \in \mathbb{R}^n$ such that $Hx = \|x\|_2 e_1$, where $H = I - \beta vv^*$

1: **function** $[\beta, v]$ = **house**(x)
2: $n = \mathtt{length}(x)$ (here $\mathtt{length}(x)$ denotes the dimension of x)
3: $\sigma = x_2^2 + x_3^2 + \cdots + x_n^2$
4: $v = x$
5: **if** $\sigma = 0$ **then**
6: $\quad \beta = 0$
7: $\quad$ **if** $x_1 < 0$ **then**
8: $\qquad v_1 = -x_1$
9: $\quad$ **end if**
10: **else**
11: $\quad v = x$
12: $\quad \mu = \sqrt{x_1^2 + \sigma}$
13: $\quad$ **if** $x_1 < 0$ **then**
14: $\qquad v_1 = x_1 - \mu$
15: $\quad$ **else**
16: $\qquad v_1 = -\sigma/(x_1 + \mu)$
17: $\quad$ **end if**
18: $\quad \beta = 2/(v_1^2 + \sigma)$
19: **end if**

☞ 在实际计算时, 我们可以将向量 v 单位化, 使得 $v_1 = 1$. 这样, 我们就无需为 v 另外分配空间, 而是将 $v(2:n)$ 存放在 $x(2:n)$ 中, 因为经过 Householder 变换后, 向量 x 除第一个分量外, 其它都为零. 同时, 为了避免可能产生的溢出, 我们也可以事先将 x 单位化, 即令 $x = x/\|x\|_2$

可以证明, 上述算法具有很好的数值稳定性, 即 [43]

$$\|\tilde{H} - H\|_2 = \mathcal{O}(\varepsilon_m),$$

其中 $\tilde{H}$ 是由上述算法计算得到的数值解, ε_m 是机器精度.

设 $A \in \mathbb{R}^{m\times n}$, $H = I - \beta vv^* \in \mathbb{R}^m$, 则

$$HA = (I - \beta vv^*)A = A - \beta vv^*A = A - \beta v(A^*v)^*.$$

因此, 在做 Householder 变换时, 并不需要生成 Householder 矩阵, 只需要 Householder 向量即可. 上面矩阵相乘的总运算量大约为 $4mn$.

3.6.4 Givens 变换

为简单起见,我们这里这讨论实数域中的 Givens 变换. 设 $\theta \in [0, 2\pi]$, 我们称矩阵

$$G(i,j,\theta) = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}$$

为 **Givens 变换** (或 **Givens 旋转**, 或 **Givens 矩阵**), 其中 $c = \cos(\theta)$, $s = \sin(\theta)$. 即将单位矩阵的 (i,i) 和 (j,j) 位置上的元素用 c 代替, 而 (i,j) 和 (j,i) 位置上的元素分别用 s 和 $-s$ 代替, 所得到的矩阵就是 $G(i,j,\theta)$.

定理 3.19 $G(i,j,\theta)$ 是正交矩阵, 且 $\det(G(i,j,\theta)) = 1$.

我们需要注意的是, 当一个矩阵左乘一个 Givens 矩阵时, 只会影响其第 i 行和第 j 行的元素. 而当一个矩阵右乘一个 Givens 矩阵时, 只会影响其第 i 和第 j 列的元素.

例 3.1 设 $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$, 则存在一个 Givens 变换 $G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \in \mathbb{R}^{2\times 2}$ 使得 $Gx = \begin{bmatrix} r \\ 0 \end{bmatrix}$, 其中 c, s 和 r 的值如下:

- 若 $x_1 = x_2 = 0$, 则 $c = 1, s = 0, r = 0$;
- 若 $x_1 = 0$ 但 $x_2 \neq 0$, 则 $c = 0, s = x_2/|x_2|, r = |x_2|$;
- 若 $x_1 \neq 0$ 但 $x_2 = 0$, 则 $c = \mathrm{sign}(x_1), s = 0, r = |x_1|$;
- 若 $x_1 \neq 0$ 且 $x_2 \neq 0$, 则 $c = x_1/r, s = x_2/r, r = \sqrt{x_1^2 + x_2^2}$.

也就是说, 通过 Givens 变换, 我们可以将向量 $x \in \mathbb{R}^2$ 的第二个分量化为 0.

事实上, 对于任意一个向量 $x \in \mathbb{R}^n$, 我们都可以通过 Givens 变换将其任意一个位置上的分量化为 0. 更进一步, 我们也可以通过若干个 Givens 变换, 将 x 中除第一个分量外的所有元素都化为 0.

算法 3.3 Givens 变换

```
   % Given x = [a, b]^T ∈ R^2, compute c and s such that Gx = [r, 0]^T where r = ||x||_2
1: function [c, s] = givens(a, b)
2: if b = 0 then
3:     if a ≥ 0 then
4:         c = 1,   s = 0
5:     else
6:         c = −1,   s = 0
7:     end if
8: else
9:     if |b| > |a| then
```

10: $\tau = \dfrac{a}{b}, \quad s = \dfrac{\text{sign}(b)}{\sqrt{1+\tau^2}}, \quad c = s\tau$

11: **else**

12: $\tau = \dfrac{b}{a}, \quad c = \dfrac{\text{sign}(a)}{\sqrt{1+\tau^2}}, \quad s = c\tau$

13: **end if**

14: **end if**

3.6.5 正交矩阵舍入误差分析

引理 3.4 设 $P \in \mathbb{R}^{n\times n}$ 是一个精确的 *Householder* 或 *Givens* 变换, $\tilde{P}$ 是其浮点运算近似, 则

$$\text{fl}(\tilde{P}A) = P(A+E), \quad \text{fl}(A\tilde{P}) = (A+F)P,$$

其中 $\|E\|_2 = \mathcal{O}(\varepsilon_m) \cdot \|A\|_2$, $\|F\|_2 = \mathcal{O}(\varepsilon_m) \cdot \|A\|_2$.

这说明对一个矩阵做 Householder 变换或 Givens 变换是向后稳定的.

定理 3.20 考虑对矩阵 A 做一系列的正交变换, 则有

$$\text{fl}(\tilde{P}_k \cdots \tilde{P}_1 A \tilde{Q}_1 \cdots \tilde{Q}_k) = P_k \cdots P_1 (A+E) Q_1 \cdots Q_k,$$

其中 $\|E\|_2 = \mathcal{O}(\varepsilon_m) \cdot (k\|A\|_2)$. 这说明整个计算过程是向后稳定的.

一般地, 假设 X 是一个非奇异的线性变换, $\tilde{X}$ 是其浮点运算近似. 当 X 作用到 A 上时, 我们有

$$\text{fl}(\tilde{X}A) = XA + E = X(A + X^{-1}E) \triangleq X(A+F),$$

其中 $\|E\|_2 = O(\varepsilon_m) \cdot \|XA\|_2 \leq O(\varepsilon_m) \cdot \|X\|_2 \cdot \|A\|_2$, 故

$$\|F\|_2 = \|X^{-1}E\|_2 \leq \mathcal{O}(\varepsilon_m) \cdot \|X^{-1}\|_2 \cdot \|X\|_2 \cdot \|A\|_2 = \mathcal{O}(\varepsilon_m) \cdot \kappa_2(X) \cdot \|A\|_2,$$

因此, 舍入误差将被放大 $\kappa_2(X)$ 倍. 当 X 是正交变换时, $\kappa_2(X)$ 达到最小值 1, 这就是为什么在浮点运算中尽量使用正交变换的原因.

3.7 QR 分解的实现

这里我们给出 QR 分解的具体实现方法. 下面我们分别介绍基于 MGS 过程, Householder 变换和 Givens 变换的 QR 分解算法.

3.7.1 基于 MGS 的 QR 分解

在证明 QR 分解的存在性时, 我们利用了 Gram-Schmidt 正交化过程. 但由于数值稳定性方面的原因, 在实际计算中, 我们一般不采用 Gram-Schmidt 过程, 取而代之的是 **修正的** Gram-Schmidt **过程** (modified Gram-Schmidt process), 即 MGS . 本算法的运算量大约为 $2mn^2$.

算法 3.4 基于 MGS 的 QR 分解

% Given $A \in \mathbb{R}^{m\times n}$, compute $Q = [q_1, \ldots, q_n] \in \mathbb{R}^{m\times n}$ and $R \in \mathbb{R}^{n\times n}$ such that $A = QR$

1: Set $R = [r_{ik}] = 0_{n\times n}$ (the $n \times n$ zero matrix)
2: **if** $a_1 = 0$ **then**
3:     $q_1 = 0$
4: **else**
5:     $r_{11} = \|a_1\|_2$
6:     $q_1 = a_1/\|a_1\|_2$
7: **end if**
8: **for** $k = 2$ to n **do**
9:     $q_k = a_k$
10:     **for** $i = 1$ to $k - 1$ **do**
11:         $r_{ik} = q_i^T q_k$
12:         $q_k = q_k - r_{ik} q_i$
13:     **end for**
14:     **if** $q_k \neq 0$ **then**
15:         $r_{kk} = \|q_k\|_2$
16:         $q_k = q_k/r_{kk}$
17:     **end if**
18: **end for**

3.7.2 基于 Householder 变换的 QR 分解

由定理 3.18 可知, 通过 Householder 变换, 我们可以将任何一个非零变量 $x \in \mathbb{R}^n$ 转化成 $\|x\|_2 e_1$, 即除第一个元素外, 其它都为零. 下面我们就考虑通过 Householder 变换来实现矩阵的 QR 分解.

我们首先考虑 $m = n$ 时的情形. 设矩阵 $A \in \mathbb{R}^{n\times n}$, 令 $H_1 \in \mathbb{R}^{n\times n}$ 为一个 Householder 变换, 满足

$$H_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} r_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

于是

$$H_1 A = \left[\begin{array}{c|ccc} r_1 & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ \hline 0 & & & \\ \vdots & & \tilde{A}_2 & \\ 0 & & & \end{array}\right],$$

其中 $\tilde{A}_2 \in \mathbb{R}^{(n-1)\times(n-1)}$. 同样地, 我们可以构造一个 Householder 变换 $\tilde{H}_2 \in \mathbb{R}^{(n-1)\times(n-1)}$, 将 $\tilde{A}_2$ 的第一列中除第一个元素外的所有元素都化为 0, 即

$$\tilde{H}_2 \tilde{A}_2 = \left[\begin{array}{c|ccc} r_2 & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ \hline 0 & & & \\ \vdots & & \tilde{A}_3 & \\ 0 & & & \end{array}\right].$$

令

$$H_2 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{H}_2 \end{bmatrix}.$$

则 $H_2 \in \mathbb{R}^{n\times n}$, 且

$$H_2 H_1 A = \left[\begin{array}{cc|ccc} r_1 & \tilde{a}_{12} & \tilde{a}_{13} & \cdots & \tilde{a}_{1n} \\ 0 & r_2 & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ \hline 0 & 0 & & & \\ \vdots & \vdots & & \tilde{A}_3 & \\ 0 & 0 & & & \end{array}\right].$$

不断重复上述过程. 这样, 我们就得到一系列的矩阵

$$H_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \tilde{H}_k \end{bmatrix}, \quad k = 1, 2, 3, 4, \ldots, n-1$$

使得

$$H_{n-1} \cdots H_2 H_1 A = \begin{bmatrix} r_1 & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ 0 & r_2 & \cdots & \tilde{a}_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_n \end{bmatrix} \triangleq R.$$

由于 Householder 变换都是正交矩阵, 因此 $H_1, H_2, \ldots, H_{n-1}$ 也都是正交矩阵. 令

$$Q = (H_{n-1} \cdots H_2 H_1)^{-1} = H_1^{-1} H_2^{-1} \cdots H_{n-1}^{-1} = H_1 H_2 \cdots H_{n-1},$$

则 Q 也是正交矩阵, 且

$$A = (H_{n-1} \cdots H_2 H_1)^{-1} R = QR.$$

以上就是基于 Householder 变换的 QR 分解的具体实现过程. 最后所得到的上三角矩阵 R 就存放在 A 的上三角部分. 矩阵 Q 可通过下面的算法实现

$$\begin{cases} Q = I_n, \\ Q = QH_k, \quad k = 1, 2, \ldots, n-1. \end{cases}$$

☞ 这里我们假定 A 是满秩的, 如果 A 不是满秩, 则可以考虑列主元 QR 分解.

☞ 如果 $m > n$, 我们仍然可以通过上面的过程进行 QR 分解, 只是最后我们得到一个正交矩阵 $Q \in \mathbb{R}^{m\times m}$ 和一个上三角矩阵 $R \in \mathbb{R}^{m\times n}$, 使得 $A = QR$.

☞ 如果不需要生成 Q, 则基于 Householder 变换的 QR 分解的总运算量大约为 $2mn^2 - 2/3n^3$.

如果保留了每一步的 Householder 向量, 则 Q 也可以通过下面的向后累积方法实现:

$$
\begin{aligned}
&Q = I_n, \\
&Q = H_k Q, \quad k = n-1, n-2, \ldots, 1.
\end{aligned}
$$

这样做的好处是一开始 Q 会比较稀疏, 随着迭代的进行, Q 才会慢慢变满. 而前面的计算方法, 第一步就将 Q 变成了一个满矩阵. 采用这种方法计算 Q 的运算量大约为 $4(m^2n - mn^2 + \frac{1}{3}n^3)$.

如果将 Q 写成下面的形式

$$
Q = I + WY^T,
$$

则可以采用分块形式来计算 W 和 Y, 虽然运算量会稍有增长, 但大多数运算是矩阵乘法, 因此可以尽可能多地采用 3 级 BLAS 运算, 效率可能会更高. 详情可参见 Golub [16]

算法 3.5 基于 Householder 变换的 QR 分解

% Given $A \in \mathbb{R}^{m\times n}$, compute Q and R such that $A = QR$ where $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{m\times n}$

% The upper triangular part of R is stored in the upper triangular part of A

1: Set $Q = I_{m\times m}$
2: **for** $k = 1$ to n **do**
3: $\quad x = A(k:m, k)$
4: $\quad [\beta, v_k] =$ **house**(x)
5: $\quad v_k = v_k / \|v_k\|_2$
6: $\quad A(k:m, k:n) = (I_{m-k+1} - 2v_k v_k^T) A(k:m, k:n)$
7: $\quad Q(1:k-1, k:m) = Q(1:k-1, k:m)(I_{m-k+1} - 2v_k v_k^T)$
8: $\quad Q(k:m, k:m) = Q(k:m, k:m)(I_{m-k+1} - 2v_k v_k^T)$
9: **end for**

☞ 上面的算法只是关于利用 Householder 变换来实现 QR 分解的一个简单描述, 并没有考虑运算量问题. 在实际计算时, 我们通常会保留所有的 Householder 向量. 由于第 k 步中 $\tilde{H}_k$ 所对应的 Householder 向量 v_k 的长度为 $m-k+1$, 因此我们先把 v_k 单位化, 使得 v_k 的第一元素为 1, 这样就只要存储 $v_k(2:end)$, 共 $m-k$ 个元素, 我们可以把这些元素存放在 A 的严格下三角部分. A 的上三角部分仍然存放 R. 在计算 Q 时采用向后累积的方法, 所以总的运算量大约为 $4m^2n - 2mn^2 + \frac{2}{3}n^3$. 若 $m = n$, 则运算量大约为 $\frac{8}{3}n^3$.

☞ 我们也可以考虑分块 Householder QR 分解, 以便充分利用 3 级 BLAS 运算, 提高计算效率.

3.7.3 列主元 QR 分解

当 A 不是满秩时, 我们可以进行**列主元 QR 分解**.

定理 3.21 (列主元 QR 分解) 设 $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$, 且 $\mathrm{rank}(A) = k < n$. 则存在置换矩阵 P, 正交矩阵 $Q \in \mathbb{C}^{m\times m}$, 使得

$$AP = Q \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}_{m\times n},$$

其中 $R_{11} \in \mathbb{C}^{k\times k}$ 是非奇异上三角矩阵, 且对角线元素满足 $r_{11} \geq r_{22} \geq \cdots \geq r_{kk} > 0$.

列主元 QR 分解的实现过程与 QR 分解基本类似, 只是在第 l 步时, 需要选个列主元, 同时可能需要做一个列交换.

假设经过 $l-1$ 步后, 我们得到下面的分解

$$AP^{(l-1)} = Q^{(l-1)} \begin{bmatrix} R_{11}^{(l-1)} & R_{12}^{(l-1)} \\ 0 & R_{22}^{(l-1)} \end{bmatrix} \triangleq Q^{(l-1)}R^{(l-1)}, \quad \text{即} \quad \left(Q^{(l-1)}\right)^T AP^{(l-1)} = R^{(l-1)},$$

其中 $P^{(l-1)}$ 是置换矩阵, $Q^{(l-1)}$ 是正交矩阵, $R_{12}^{(l-1)} \in \mathbb{R}^{(l-1)\times(l-1)}$ 是非奇异上三角矩阵.

下面进行第 l 步:

(1) 首先计算 $R_{22}^{(l-1)}$ 的所有列的范数, 如果范数都为 0, 则 $R_{22}^{(l-1)} = 0$, 此时必有 $l-1 = k$, 算法结束.

(2) 当 $l \leq k$ 时, $R_{22}^{(l-1)} \neq 0$, 记范数最大的列为第 i_l 列 (如果有相等的, 取其中一个即可). 若 $i_l \neq 1$, 则交换 $R^{(l-1)}$ 的第 l 列与第 $i_l + l - 1$ 列, 并记相应的置换矩阵为 P_l.

(3) 以 $R_{22}^{(l-1)}$ 的第 i_l 列为 Householder 向量, 构造 Householder 变换 $\tilde{H}_l$, 并令 $H_l = \begin{bmatrix} I_{l-1} & 0 \\ 0 & \tilde{H}_l \end{bmatrix}$, $P^{(l)} = P^{(l-1)}P_l$, 则

$$H_l \left(Q^{(l-1)}\right)^T AP^{(l)} = H_l R^{(l-1)} P_l = \begin{bmatrix} R_{11}^{(l-1)} & \tilde{R}_{12}^{(l-1)} \\ 0 & \tilde{R}_{22}^{(l-1)} \end{bmatrix} \triangleq R^{(l)},$$

其中 $\tilde{R}_{22}^{(l-1)}$ 的第一列除第一个元素外, 其余都是零. 且该元素的值等于 $R_{22}^{(l-1)}$ 第 i_l 列的范数. 记 $Q^{(l)} \triangleq Q^{(l-1)}H_l^T$, 则

$$AP^{(l)} = Q^{(l)}R^{(l)} = \begin{bmatrix} R_{11}^{(l)} & R_{12}^{(l)} \\ 0 & R_{22}^{(l)} \end{bmatrix},$$

其中 $R_{11}^{(l)} \in \mathbb{R}^{l\times l}$ 为非奇异上三角矩阵.

依此类推, 直到第 k 步, 我们就可以得到 A 的列主元 QR 分解, 其中 R_{11} 的对角线元素的递减关系是由于选取列主元和 Householder 变换的性质得到的.

3.7.4 基于 Givens 变换的 QR 分解

我们同样可以利用 Givens 变换来做 QR 分解.

设 $A \in \mathbb{R}^{n\times n}$, 首先构造一个 Givens 变换 G_{21}, 作用在 A 的最前面的两行上, 使得

$$G_{21} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} \tilde{a}_{11} \\ 0 \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix}.$$

由于 G_{21} 只改变矩阵的第 1 行和第 2 行的值, 所以其它行保存不变. 然后再构造一个 Givens 变换 G_{31}, 作用在 $G_{21}A$ 的第 1 行和第 3 行, 将第三个元素化为零. 由于 G_{31} 只改变矩阵的第 1 行和第 3 行的值, 所以第二行的零元素维持不变. 以此类推, 我们可以构造一系列的 Givens 变换 $G_{41}, G_{51}, \ldots, G_{n1}$, 使得 $G_{n1}\cdots G_{21}A$ 的第一列中除第一个元素外, 其它元素都化为零, 即

$$G_{n1}\cdots G_{21}A = \begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \cdots & * \end{bmatrix}.$$

下面我们可以对第二列进行类似的处理. 构造 Givens 变换 $G_{32}, G_{42}, \ldots, G_{n2}$, 第二列的第 3 至第 n 个元素全化为零, 同时保持第一列不变.

以此类推, 我们对其他列也做类似的处理. 最后, 通过构造 $\frac{1}{2}n(n-1)$ 个 Givens 变换, 将 A 转化成一个上三角矩阵 R, 即

$$R = G_{n,n-1}\cdots G_{21}A.$$

令 $Q = (G_{n,n-1}\cdots G_{21})^T$. 由于 Givens 变换是正交矩阵, 所以 Q 也是正交矩阵. 于是, 我们就得到矩阵 A 的 QR 分解

$$A = QR.$$

与 Householder 变换一样, 在进行 Givens 变换时, 我们不需要显式地写出 Givens 矩阵.

对于稠密矩阵而言, 基于 Givens 变换的 QR 分解的运算量比 Householder 变换要多很多. 当需要连续应用一系列 Givens 变换时, 我们可以使用快速 Givens 变换 (见 [6]), 但即便如此, 其速度仍然要慢于 Householder 变换, 因此基于 Givens 变换的 QR 分解主要用于当矩阵的非零下三角元素相对较少时的情形, 比如对上 Hessenberg 矩阵进行 QR 分解.

☞ 如果 $A \in \mathbb{R}^{m\times n}$, 其中 $m > n$, 我们仍然可以通过 Givens 变换进行 QR 分解.

下面是关于 Givens QR 分解的一个简单描述.

算法 3.6 基于 Givens 变换的 QR 分解

% Given $A \in \mathbb{R}^{m\times n}$, compute Q and R such that $A = QR$ where $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{m\times n}$
% The upper triangular part of R is stored in the upper triangular part of A

1: Set $Q = I_{m\times m}$
2: **for** $k = 1$ to n **do**
3:     **for** $i = k+1$ to m **do**
4:         $[c, s]$=**givens**(a_{kk}, a_{ik})
5:         $\begin{bmatrix} A(k,k:n) \\ A(i,k:n) \end{bmatrix} = G \begin{bmatrix} A(k,k:n) \\ A(i,k:n) \end{bmatrix}$ where $G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$
6:         $[Q(1:m,k), Q(1:m,i)] = [Q(1:m,k), Q(1:m,i)]G^T$
7:     **end for**
8: **end for**

3.7.5 QR 分解的稳定性

基于 Householder 变换和 Givens 变换的 QR 分解都具有很好的数值稳定性. 详细分析可以参考 [43] 和 [21]. 基于 MGS 的 QR 分解也是向后稳定的, 参见 [28].

当需要计算矩阵 Q 时, 基于 MGS 的 QR 分解的运算量相对较少, 因此当 A 的列向量具有很好的线性无关性时, 我们可以使用 MGS 来计算 QR 分解.

但是, 由于舍入误差的原因, 最后得到的矩阵 Q 会带有一定的误差, 可能会导致 Q 失去正交性.

Björck [5] 证明了, 通过 MGS 计算的矩阵 Q 满足

$$Q^T Q = I + E_{MGS} \quad \text{其中} \quad \|E_{MGS}\|_2 \approx \varepsilon_m \kappa_2(A).$$

而 Householder 变换计算的矩阵 Q 满足

$$Q^T Q = I + E_H \quad \text{其中} \quad \|E_H\|_2 \approx \varepsilon_m.$$

因此, 如果正交性至关重要, 则当 A 的列向量接近线性相关时, 最好使用 Householder 变换.

3.8 广义逆与最小二乘

广义逆的概念最早由 Moore [27] 于 1920 年提出, 他给出的定义如下: 设 $A \in \mathbb{C}^{m\times n}$, 若 $X \in \mathbb{C}^{n\times m}$ 满足

$$AX = P_{\mathrm{Ran}(A)}, \quad XA = P_{\mathrm{Ran}(X)},$$

即 AX 和 XA 分别为 $\mathrm{Ran}(A)$ 和 $\mathrm{Ran}(X)$ 上的正交投影算子, 则称 X 是 A 的广义逆.

1955 年, Penrose [31] 利用下面四个矩阵方程给出了广义逆的定义.

定义 3.3 设 $A \in \mathbb{C}^{m\times n}$, 若 $X \in \mathbb{C}^{n\times m}$ 满足

$$AXA = A \tag{3.19}$$

$$XAX = X \tag{3.20}$$

$$(AX)^* = AX \tag{3.21}$$

$$(XA)^* = XA. \tag{3.22}$$

则称 X 为 A 的**广义逆** (或 Moore-Penrose 逆, 简称 MP 逆), 记为 $A^\dagger$.

可以证明, 以上两个定义是等价的.

定理 3.22 [51] 设 $A \in \mathbb{C}^{m\times n}$, 则满足矩阵方程 (3.19)-(3.22) 的矩阵 $X \in \mathbb{C}^{n\times m}$ 存在且唯一.

☞ 若 $A \in \mathbb{C}^{n\times n}$ 非奇异, 则 $A^\dagger = A^{-1}$.

设 $A \in \mathbb{C}^{m\times n}$, $\mathrm{rank}(A) = r > 0$, A 的 SVD 为

$$A = U \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} V^*, \quad \Sigma_1 = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}.$$

则容易验证

$$A^\dagger = V \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^*.$$

下面是广义逆的一些基本性质.

定理 3.23 设 $A \in \mathbb{C}^{m\times n}$, 则

(1) $(A^\dagger)^\dagger = A$;

(2) $(A^T)^\dagger = (A^\dagger)^T$, $(A^*)^\dagger = (A^\dagger)^*$;

(3) $\mathrm{rank}(A) = \mathrm{rank}(A^\dagger) = \mathrm{rank}(A^\dagger A)$;

(4) $(AA^*)^\dagger = (A^*)^\dagger A^\dagger$, $(A^*A)^\dagger = A^\dagger (A^*)^\dagger$;

(5) $(AA^*)^\dagger AA^* = AA^\dagger$, $(A^*A)^\dagger A^*A = A^\dagger A$;

(6) $A^\dagger = (A^*A)^\dagger A^* = A^*(AA^*)^\dagger$,
特别地, 若 A 列满秩, 则 $A^\dagger = (A^*A)^{-1}A^*$, 若 A 行满秩, 则 $A^\dagger = A^*(AA^*)^{-1}$;

(7) 若 U, V 是酉矩阵, 则 $(UAV)^\dagger = V^*A^\dagger U^*$.

☞ 一般来说, 当 A, B 是方阵时,

- $(AB)^\dagger \neq B^\dagger A^\dagger$;
- $AA^\dagger \neq A^\dagger A$;
- $(A^k)^\dagger \neq (A^\dagger)^k$;
- A 和 $A^\dagger$ 的非零特征值并不是互为倒数.

定理 3.24 设 $A \in \mathbb{C}^{m\times n}$, 则

$$\begin{aligned}
&\mathrm{Ran}(AA^\dagger) = \mathrm{Ran}(AA^*) = \mathrm{Ran}(A),\\
&\mathrm{Ran}(A^\dagger A) = \mathrm{Ran}(A^*A) = \mathrm{Ran}(A^*) = \mathrm{Ran}(A^\dagger),\\
&\mathrm{Ker}(AA^\dagger) = \mathrm{Ker}(AA^*) = \mathrm{Ker}(A^*) = \mathrm{Ker}(A^\dagger),\\
&\mathrm{Ker}(A^\dagger A) = \mathrm{Ker}(A^*A) = \mathrm{Ker}(A).
\end{aligned}$$

推论 3.25 (广义逆与正交投影) 设 $A \in \mathbb{C}^{m\times n}$, 则

$$P_A = AA^\dagger, \quad P_{A^T} = A^\dagger A,$$

其中 P_A 和 P_{A^T} 分别表示 $\mathrm{Ran}(A)$ 和 $\mathrm{Ran}(A^T)$ 上的正交投影变换.

3.8.1 广义逆的计算

利用 SVD

我们可以利用 A 的奇异值分解来计算 $A^\dagger$. 但是一般地讲, 这个工作量很大. 因为奇异值分解的工作量与求 AA^* 或 A^*A 的特征值问题有关.

利用满秩分解

To be continued ...

Greville 递推算法

To be continued ...

3.8.2 广义逆与线性最小二乘

定理 3.26 设 $A \in \mathbb{R}^{m\times n}$, 则线性最小二乘问题 (3.1) 的解为

$$x = A^\dagger b + (I - P_{A^T})z, \quad \forall\, z \in \mathbb{R}^n. \tag{3.23}$$

证明. 参见 [51, page 85]. □

☞ 通常, 线性最小二乘问题的解 (3.23) 不是唯一的. 但当 A 列满秩时, $P_{A^T} = I$, 此时解唯一.

定理 3.27 设 $A \in \mathbb{R}^{m\times n}$ 的解集为 $\mathcal{S}$, 则

$$\min_{x\in\mathcal{S}} \|x\|_2 \tag{3.24}$$

存在唯一解, 即满足 (3.24) 的线性最小二乘问题 (3.1) 的解存在且唯一.

证明. 参见 [51, page 85]. □

3.9 课后习题

3.1 设 $A \in \mathbb{R}^{m\times n}$, 证明: $\mathrm{Ran}(A^T) = \mathrm{Ran}(A^T A)$.

3.2 证明定理 3.7, 即奇异值的相关性质.

3.3 证明奇异值的最小最大定理, 即定理 3.10.

3.4 证明奇异值的扰动性质, 即定理 3.12 .

3.5 设矩阵 A 的奇异值分解为 $A = U_n \Sigma V^* = \sum_{i=1}^{n} \sigma_i u_i v_i^* \in \mathbb{C}^{m\times n}$, 证明

$$\min_{B\in\mathbb{C}^{m\times n},\ \mathrm{rank}(B)=k} \|A-B\|_F = \|A-A_k\|_F = \sqrt{\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2}\ .$$

其中 $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^*$.

3.6 设 $A \in \mathbb{R}^{n\times n}$, 证明 $\rho(A) \le \sigma_{max}(A)$.

3.7 设 $A \in \mathbb{R}^{n\times n}$, 证明 $\|A\|_F^2 \ge \sum_{i=1}^{n} \lambda_i^2(A)$.
(提示: 利用 F-范数与迹的关系, 以及 QR 分解)

3.8 设 $\tau \neq 0$, 向量 $u, v \in \mathbb{R}^n$ 均不为零. 求矩阵 $E(u,v,\tau) = I - \tau u v^*$ 的特征值.

3.9 设 $A \in \mathbb{R}^{n\times n}$ 是正交矩阵, 则 A 可表示成至多 $n-1$ 个 Householder 变换的乘积.

3.10 设 $A \in \mathbb{R}^{n\times n}$ 是正交矩阵, 则 A 可表示成至多 $\frac{1}{2}n(n-1)$ 个 Givens 变换的乘积.

3.11 设 $x = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$ 是一个非零向量, H 是 Householder 矩阵, 满足 $Hx = \alpha e_1$.
证明: H 的第一列与 x 平行.

3.12 设 $H_k \in \mathbb{R}^{k\times k}$ 是 Householder 变换, 其中 $k < n$.
证明: $H_n = \begin{bmatrix} I_{n-k} & 0 \\ 0 & H_k \end{bmatrix}$ 是 n 阶 Householder 变换.

3.13 (极分解) 设 $A \in \mathbb{C}^{n\times n}$.
证明: 存在酉矩阵 U 和唯一的 Hermite 半正定矩阵 P, 使得 $A = PU$.
进一步, 若 A 非奇异, 则 U 也是唯一的.

3.14 设 $A \in \mathbb{C}^{n\times n}$.
证明: A 可对角化当且仅当存在 Hermite 正定矩阵 P 使得 $P^{-1}AP$ 是正规矩阵.
(提示: 利用极分解, 但不是对 A 进行极分解)

3.15 设 $R = \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix} \in \mathbb{R}^{2\times 2}$, 其中 $r_{11} \neq r_{22}$, 求一个 Givens 变换 G, 使得 $G^T R G = \begin{bmatrix} r_{22} & r_{12} \\ 0 & r_{11} \end{bmatrix}$
.

3.16 设 $R \in \mathbb{R}^{n\times n}$ 是一个上三角矩阵, 且对角线元素互不相同.
证明: 存在正交矩阵 Q, 使得 $Q^T R Q$ 为上三角矩阵, 且对角线元素为 R 的对角线元素的降序排列.

3.17 设 $A \in \mathbb{C}^{n\times n}$ 有 n 个不同的特征值, 其 Schur 分解为 $A = URU^*$, 其中 U 为酉矩阵, R 为上三角矩阵. 设矩阵 $B \in \mathbb{C}^{n\times n}$ 满足 $AB = BA$. 证明: U^*BU 是上三角矩阵.

计算题

3.18 用 Householder 变换计算矩阵 $A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & -1 & -1 \\ 2 & -4 & 10 \end{bmatrix}$ 的 QR 分解.

3.19 设 $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ 且 $\mathrm{rank}(A) = n$. 计算矩阵 $\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix}$ 的条件数. (用 A 的奇异值表示)

3.20 设 $A \in \mathbb{R}^{n\times n}$ 是一个对角加边矩阵

$$A = \begin{bmatrix} a_1 & b_2 & b_3 & \cdots & b_n \\ c_2 & a_2 & & & \\ c_3 & & a_3 & & \\ \vdots & & & \ddots & \\ c_n & & & & a_n \end{bmatrix}.$$

试给出用 Givens 变换计算 A 的 QR 分解的详细算法.

实践题

3.21 编写基于 Householder 变换的 QR 分解的 Matlab 函数: $[Q, R]$=**QR_Householder**(A) .

3.22 设 $A = \begin{bmatrix} R \\ S \end{bmatrix}$, 其中 $R \in \mathbb{R}^{n\times n}$ 是上三角矩阵, $S \in \mathbb{R}^{m\times n}$ 是稠密矩阵, 试描述 A 的基于 Householder 变换的 QR 算法. 要求算法过程中始终保持 R 中的零元.

3.23 设 $A = R + uv^T$, 其中 $R \in \mathbb{R}^{n\times n}$ 是上三角矩阵, $u, v \in \mathbb{R}^n$ 为非零列向量, 试给出计算 A 的 QR 分解的有效算法. (提示: 使用 Givens 变换, 算法总运算量大约为 $O(n^2)$)

3.24 构造列主元 QR 分解的算法.

3.25 构造矩阵的 LQ 分解算法, 其中 L 为下三角矩阵, Q 为正交矩阵.

第四章　非对称特征值问题

设 $A \in \mathbb{R}^{n\times n}$ 是一个非对称的稠密矩阵, 本章主要讨论如何计算 A 的全部特征值和特征向量. 记 A 的特征值为 $\lambda_1, \lambda_2, \ldots, \lambda_n$. 本章中我们总是假定

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n| \geq 0,$$

即 A 的特征值按绝对值（模）降序排列.

本章主要介绍以下方法:

- 幂迭代算法
- 位移策略与反迭代技巧
- 正交迭代法
- QR 算法
- Hessenberg 矩阵与实用的 QR 算法

关于稠密矩阵特征值计算的参考资料有:

- J. H. Wilkinson, The Algebraic Eigenvalue Problem, 1965 [43] (有中文翻译)
- B. N. Parlett, The Symmetric Eigenvalue Problem, 2nd Eds., 1998 [29]
- G. W. Stewart, Matrix Algorithms, Vol II: Eigensystems, 2001 [37]

4.1 幂迭代

4.1.1 幂迭代算法

幂迭代是计算特征值和特征向量的一种简单易用的算法. 幂迭代虽然简单, 但它却建立了计算特征值和特征向量的算法的一个基本框架.

算法 4.1 幂迭代算法 (Power Iteration)

1: Choose an initial guess $x^{(0)}$ with $\|x^{(0)}\|_2 = 1$
2: set $k = 0$
3: **while** not convergence **do**
4: $\quad y^{(k+1)} = Ax^{(k)}$
5: $\quad x^{(k+1)} = y^{(k+1)}/\|y^{(k+1)}\|_2$
6: $\quad \mu_{k+1} = (x^{(k+1)}, Ax^{(k+1)})$ % 内积
7: $\quad k = k + 1$
8: **end while**

下面讨论幂迭代的收敛性. 假设

(1) $A \in \mathbb{R}^{n\times n}$ 是可对角化的, 即 $A = V\Lambda V^{-1}$, 其中 $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \in \mathbb{C}^{n\times n}$, $V = [v_1, v_2, \dots, v_n] \in \mathbb{C}^{n\times n}$, 且 $\|v_i\|_2 = 1$ $(i = 1, 2, \dots, n)$.

(2) 同时, 我们还假设 $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$.

由于 $V \in \mathbb{C}^{n\times n}$ 非奇异, 所以它的列向量组构成 $\mathbb{C}^n$ 的一组基. 因此迭代初始向量 $x^{(0)}$ 可表示为

$$x^{(0)} = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n = V[\alpha_1, \alpha_2, \dots, \alpha_n]^T.$$

我们假定 $\alpha_1 \neq 0$, 即 $x^{(0)}$ 不属于不变子空间 $\mathrm{span}\{v_2, v_3, \dots, v_n\}$ (由于 $x^{(0)}$ 是随机选取的, 从概率意义上讲, 这个假设通常是成立的). 于是我们可得

$$A^k x^{(0)} = (V\Lambda V^{-1})^k V \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = V\Lambda^k \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = V \begin{bmatrix} \alpha_1\lambda_1^k \\ \alpha_2\lambda_2^k \\ \vdots \\ \alpha_n\lambda_n^k \end{bmatrix} = \alpha_1\lambda_1^k V \begin{bmatrix} 1 \\ \dfrac{\alpha_2}{\alpha_1}\left(\dfrac{\lambda_2}{\lambda_1}\right)^k \\ \vdots \\ \dfrac{\alpha_n}{\alpha_1}\left(\dfrac{\lambda_n}{\lambda_1}\right)^k \end{bmatrix}.$$

又 $|\lambda_i/\lambda_1| < 1, i = 2, 3, \dots, n$, 所以

$$\lim_{k\to\infty} \left(\frac{\lambda_i}{\lambda_1}\right)^k = 0, \quad i = 2, 3, \dots, n.$$

故当 k 趋向于无穷大时, 向量

$$\left[1, \frac{\alpha_2}{\alpha_1}\left(\frac{\lambda_2}{\lambda_1}\right)^k, \dots, \frac{\alpha_n}{\alpha_1}\left(\frac{\lambda_n}{\lambda_1}\right)^k\right]^T, \quad k = 0, 1, 2, \dots$$

收敛到 $e_1 = [1, 0, \dots, 0]^T$. 所以向量 $x^{(k)} = A^k x^{(0)}/\|A^k x^{(0)}\|_2$ 收敛到 $\pm v_1$, 即 A 的对应于 (模) 最大的特征值 λ_1 的特征向量. 而 $\mu_k = (x^{(k)})^* Ax^{(k)}$ 则收敛到 $v_1^* A v_1 = \lambda_1$.

显然, 幂迭代的收敛快慢取决于 $|\lambda_2/\lambda_1|$ 的大小, $|\lambda_2/\lambda_1|$ 越小, 收敛越快.

通过上面的分析可知, 幂迭代只能用于计算矩阵的 (模) 最大的特征值和其相应的特征向量. 当 $|\lambda_2/\lambda_1|$ 接近于 1 时, 收敛速度会非常慢. 同时, 如果 A 的模最大的特征值如果是一对共轭复数, 则幂迭代就可能就会失效.

4.1.2 位移策略

为了加快幂迭代算法的收敛速度, 我们需要尽可能地减小 $|\lambda_2/\lambda_1|$ 的值. 一个简单易用的策略就是使用位移策略, 即计算矩阵 $A-\sigma I$ 的特征值, 其中 σ 是一个给定的数. 我们称 σ 为位移 (shift), 其取值满足

(1) $\lambda_1-\sigma$ 是 $A-\sigma I$ 的模最大的特征值;

(2) $\max\limits_{2\le i\le n}\left|\dfrac{\lambda_i-\sigma}{\lambda_1-\sigma}\right|$ 尽可能地小.

其中第一个条件保证最后所求得的特征值是我们所要的, 第二个条件用于加快幂迭代的收敛速度. 位移策略在特征值计算中经常被使用, 是一个非常有效的加速方法.

4.1.3 反迭代算法

如果我们使用幂迭代求 A^{-1} 的特征值, 则可求出 A 的模最小的特征值. 事实上, 利用这种思想和位移策略, 我们就可以计算矩阵的任意一个特征值.

算法 4.2 反迭代算法 (Inverse Iteration)

1: Choose a scalar σ and an initial vector $x^{(0)}$ with $\|x^{(0)}\|_2=1$
2: set $k=0$
3: **while** not convergence **do**
4: $\quad y^{(k+1)}=(A-\sigma I)^{-1}x^{(k)}$
5: $\quad x^{(k+1)}=y^{(k+1)}/\|y^{(k+1)}\|_2$
6: $\quad \mu_{k+1}=(x^{(k+1)},Ax^{(k+1)})$
7: $\quad k=k+1$
8: **end while**

这个算法称为反迭代. 在该算法中, μ_k 收敛到距离 σ 最近的 A 的特征值, 而 $x^{(k)}$ 则收敛到其对应的特征向量.

若选取离 λ_k 非常接近的位移 σ, 则与 $(A-\sigma I)^{-1}$ 的其它特征值相比, $(\lambda_k-\sigma)^{-1}$ 的值可以是非常大, 因此反迭代算法的收敛速度是非常快的.

反迭代算法的另一个优点是, 只要选取合适的位移 σ, 就可以计算 A 的任意一个特征值.

但反迭代算法的缺点也很明显: 每步迭代需要解一个线性方程组 $(A-\sigma I)y^{(k+1)}=x^{(k)}$, 这就需要对 $A-\sigma I$ 做一次 LU 分解. 而且, 与幂迭代一样, 反迭代算法一次只能求一个特征值.

4.1.4 Rayleigh 商迭代

在反迭代算法中, 有一个很重要的问题需要考虑, 即位移 σ 如何选? ?

显然, 选取 σ 的基本原则是使得其与所求的特征值越靠近越好. 这里需要指出的是, 在每步迭代中, 位移 σ 可以不一样, 即在迭代过程中可以选取不同的位移 σ.

由于在反迭代算法 4.2 中, μ_k 是收敛到所求的特征值的, 所以我们可以选取 μ_k 作为第 k 步的位移. 这时所得到的反迭代算法就称为 Rayleigh 商迭代 (Rayleigh Quotient Iteration), 简记为 RQI.

算法 4.3 Rayleigh 商迭代算法 (Rayleigh Quotient Iteration (RQI))

1: Choose an initial vector $x^{(0)}$ with $\|x^{(0)}\|_2 = 1$
2: set $k = 0$
3: compute $\sigma = (x^{(0)})^* A x^{(0)}$
4: **while** not converge **do**
5: $\quad y^{(k+1)} = (A - \sigma I)^{-1} x^{(k)}$
6: $\quad x^{(k+1)} = y^{(k+1)} / \|y^{(k+1)}\|_2$
7: $\quad \mu_{k+1} = (x^{(k+1)}, A x^{(k+1)})$
8: $\quad \sigma = \mu_{k+1}$
9: $\quad k = k + 1$
10: **end while**

下面我们讨论 Rayleigh 商迭代算法的收敛性. 首先给出收敛速度的定义.

定义 4.1 设点列 $\{\varepsilon_k\}_{k=1}^{\infty}$ 收敛, 且 $\lim\limits_{k=\infty} \varepsilon_k = 0$. 若存在一个有界常数 $0 < c < \infty$, 使得

$$\lim_{k \to \infty} \frac{|\varepsilon_{k+1}|}{|\varepsilon_k|^p} = c,$$

则称点列 $\{\varepsilon_k\}$ 是 p 次 (渐进) 收敛的. 若 $1 < p < 2$ 或 $p = 1$ 且 $c = 0$, 则称点列是超线性收敛的.

关于 RQI 算法的收敛性, 我们有下面的定理.

定理 4.1 若所求的特征值是单重的, 则当第 k 步误差充分小时, 第 $k+1$ 步误差是第 k 步误差的三次方, 即 RQI 算法是局部三次收敛的.

证明. *To be continued ...* □

关于 RQI 算法的全局收敛性, 可见参考文献 [29].

4.2 正交迭代

幂迭代和反迭代都只能同时计算一个特征对. 如果想同时计算多个特征对, 我们可以采用多个初始向量进行迭代. 而正交迭代算法就是基于这种思想, 它能够计算 A 的一个不变子空间, 从而可以同时计算出多个特征值.

算法 4.4 正交迭代算法 (Orthogonal Iteration)

1: Choose an $n \times p$ column orthogonal matrix Z_0
2: set $k = 0$
3: **while** not convergence **do**
4: compute $Y_{k+1} = AZ_k$
5: $Y_{k+1} = Z_{k+1}\hat{R}_{k+1}$ % QR 分解
6: $k = k + 1$
7: **end while**

在算法中使用 QR 分解是为了保持 Z_k 的列正交性, 使得其列向量组构成子空间 $\text{span}\{A^i Z_0\}$ 的一组正交基, 以确保算法的数值稳定性.

下面我们分析该算法的收敛性质. 假设 A 是可对角化的, 即 $A = V\Lambda V^{-1}$, 其中 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, 且 $|\lambda_1| \geq \cdots \geq |\lambda_p| > |\lambda_{p+1}| \geq \cdots \geq |\lambda_n|$. 则可得

$$\text{span}\{Z_k\} = \text{span}\{Y_k\} = \text{span}\{AZ_{k-1}\}, \quad k = 1, 2, \ldots,$$

由此可知

$$\text{span}\{Z_k\} = \text{span}\{A^k Z_0\} = \text{span}\{V\Lambda^k V^{-1} Z_0\}.$$

我们注意到

$$\Lambda^k V^{-1} Z_0 = \lambda_p^k \begin{bmatrix} (\lambda_1/\lambda_p)^k & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & (\lambda_n/\lambda_p)^k \end{bmatrix} V^{-1} Z_0 \triangleq \lambda_p^k \begin{bmatrix} W_p^{(k)} \\ W_{n-p}^{(k)} \end{bmatrix}.$$

由于当 $i > p$ 时有 $|\lambda_i/\lambda_p| < 1$, 所以当 k 趋于无穷大时, $W_{n-p}^{(k)}$ 趋向于 0. 令 $V = [V_p, V_{n-p}]$, 则

$$V\Lambda^k V^{-1} Z_0 = \lambda_p^k [V_p, V_{n-p}] \begin{bmatrix} W_p^{(k)} \\ W_{n-p}^{(k)} \end{bmatrix} = \lambda_p^k \left(V_p W_p^{(k)} + V_{n-p} W_{n-p}^{(k)} \right).$$

所以当 $k \to \infty$ 时, 有

$$\begin{aligned} \text{span}\{Z_k\} = \text{span}\{V\Lambda^k V^{-1} Z_0\} &= \text{span}\{V_p W_p^{(k)} + V_{n-p} W_{n-p}^{(k)}\} \\ &\to \text{span}\{V_p W_p^{(k)}\} = \text{span}\{V_p\}, \end{aligned}$$

即 $\text{span}\{Z_k\}$ 趋向于 A 的一个 p 维不变子空间 $\text{span}\{V_p\}$.

当 A 不可对角化时, 利用 Jordan 标准型, 我们可以得到同样的结论, 见 [41, 42].

在正交迭代中, 如果我们取 $Z_0 = I$, 则可得到一类特殊的正交迭代算法. 此时, 在一定条件下, 正交迭代会收敛到 A 的 Schur 标准型.

4.3 QR 迭代

4.3.1 算法介绍

在正交迭代中, 如果取 $Z_0 = I$, 则正交迭代与下面的算法等价.

算法 4.5 QR 迭代算法 (QR Iteration)

1: Set $A_1 = A$ and $k = 1$
2: **while** not convergence **do**
3: $\quad A_k = Q_k R_k \quad$ % QR 分解
4: $\quad$ compute $A_{k+1} = R_k Q_k$
5: $\quad k = k + 1$
6: **end while**

这个算法就是 QR 迭代 算法. 在该算法中, 我们有

$$A_{k+1} = R_k Q_k = (Q_k^T Q_k) R_k Q_k = Q_k^T (Q_k R_k) Q_k = Q_k^T A_k Q_k.$$

由这个递推关系可得

$$A_{k+1} = Q_k^T A_k Q_k = Q_k^T Q_{k-1}^T A_{k-1} Q_{k-1} Q_k = \cdots = Q_k^T Q_{k-1}^T \cdots Q_1^T A Q_1 \cdots Q_{k-1} Q_k.$$

记 $\tilde{Q}_k = Q_1 \cdots Q_{k-1} Q_k = \left[\tilde{q}_1^{(k)}, \tilde{q}_2^{(k)}, \ldots, \tilde{q}_n^{(k)}\right]$, 则

$$A_{k+1} = \tilde{Q}_k^T A \tilde{Q}_k, \tag{4.1}$$

即 A_{k+1} 与 A 正交相似.

4.3.2 QR 迭代与幂迭代的关系

记 $\tilde{R}_k = R_k R_{k-1} \cdots R_1$, 则有

$$\tilde{Q}_k \tilde{R}_k = \tilde{Q}_{k-1} (Q_k R_k) \tilde{R}_{k-1} = \tilde{Q}_{k-1} (A_k) \tilde{R}_{k-1} = \tilde{Q}_{k-1} (\tilde{Q}_{k-1}^T A \tilde{Q}_{k-1}) \tilde{R}_{k-1} = A \tilde{Q}_{k-1} \tilde{R}_{k-1},$$

由此递推下去, 即可得

$$\tilde{Q}_k \tilde{R}_k = A^{k-1} \tilde{Q}_1 \tilde{R}_1 = A^{k-1} Q_1 R_1 = A^k. \tag{4.2}$$

故

$$\tilde{Q}_k \tilde{R}_k e_1 = A^k e_1,$$

这说明 QR 迭代与幂迭代是有关系的.

假设 $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$, 则当 k 充分大时, $A^k e_1$ 收敛到 A 的模最大特征值 λ_1 对应的特征向量, 故 $\tilde{Q}_k$ 的第一列 $\tilde{q}_1^{(k)}$ 也收敛到 λ_1 对应的特征向量. 因此, 当 k 充分大时, $A\tilde{q}_1^{(k)} \to \lambda_1 \tilde{q}_1^{(k)}$, 此时由 (4.1) 可知, A_{k+1} 的第一列为

$$A_{k+1}(:, 1) = \tilde{Q}_k^T A \tilde{q}_1^{(k)} \to \lambda_1 \tilde{Q}_k^T \tilde{q}_1^{(k)} = \lambda_1 e_1,$$

即 A_{k+1} 的第一列的第一个元素收敛到 λ_1, 而其它元素都趋向于 0, 收敛速度取决于 $|\lambda_2/\lambda_1|$ 的大小.

4.3.3 QR 迭代与反迭代的关系

下面观察 $\tilde{Q}_k$ 的最后一列. 由 (4.1) 可知

$$A\tilde{Q}_k = \tilde{Q}_k A_{k+1} = \tilde{Q}_k Q_{k+1} R_{k+1} = \tilde{Q}_{k+1} R_{k+1},$$

所以有

$$\tilde{Q}_{k+1} = A\tilde{Q}_k R_{k+1}^{-1}.$$

由于 $\tilde{Q}_{k+1}$ 和 $\tilde{Q}_k$ 都是正交矩阵, 上式两边转置后求逆, 可得

$$\tilde{Q}_{k+1} = \left(\tilde{Q}_{k+1}^T\right)^{-1} = \left(\left(R_{k+1}^{-1}\right)^T \tilde{Q}_k^T A^T\right)^{-1} = \left(A^T\right)^{-1} \tilde{Q}_k R_{k+1}^T.$$

观察等式两边矩阵的最后一列, 可得

$$\tilde{q}_n^{(k+1)} = c_1 \left(A^T\right)^{-1} \tilde{q}_n^{(k)},$$

其中 c_1 为某个常数. 依此类推, 可知

$$\tilde{q}_n^{(k+1)} = c \left(A^T\right)^{-k} \tilde{q}_n^{(1)},$$

其中 c 为某个常数. 假设 A 的特征值满足 $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_{n-1}| > |\lambda_n| > 0$, 则 λ_n^{-1} 是 $\left(A^T\right)^{-1}$ 的模最大的特征值. 由幂迭代的收敛性可知, $\tilde{q}_n^{(k+1)}$ 收敛到 $\left(A^T\right)^{-1}$ 的模最大特征值 λ_n^{-1} 所对应的特征向量, 即当 k 充分大时, 有

$$\left(A^T\right)^{-1} \tilde{q}_n^{(k+1)} \to \lambda_n^{-1} \tilde{q}_n^{(k+1)}.$$

所以

$$A^T \tilde{q}_n^{(k+1)} \to \lambda_n \tilde{q}_n^{(k+1)}.$$

由 (4.1) 可知, A_{k+1}^T 的最后一列为

$$A_{k+1}^T(:, n) = \tilde{Q}_k^T A^T \tilde{q}_n^{(k)} \to \lambda_n \tilde{Q}_k^T \tilde{q}_n^{(k)} = \lambda_n e_n,$$

即 A_{k+1} 的最后一行的最后一个元素收敛到 λ_n, 而其它元素都趋向于 0, 收敛速度取决于 $|\lambda_n/\lambda_{n-1}|$ 的大小.

4.3.4 QR 迭代与正交迭代的关系

下面的定理给出了 QR 迭代算法与正交迭代算法 (取 $Z_0 = I$) 之间的关系.

定理 4.2 *设正交迭代算法 4.4 和 QR 算法 4.5 中所涉及的 QR 分解都是唯一的. A_k 是由 QR 迭代算法 4.5 生成的矩阵, Z_k 是由正交迭代算法 4.4 (取 $Z_0 = I$) 生成的矩阵, 则有*

$$A_{k+1} = Z_k^T A Z_k.$$

因此, 当 A 的所有特征值的绝对值都互不相同时, A_k 收敛到 A 的 Schur 标准型.

证明. 我们用归纳法证明该结论.

当 $k=0$ 时, $A_1 = A$, $Z_0 = I$. 结论显然成立.

设 $A_k = Z_{k-1}^T A Z_{k-1}$. 由于 Z_{k-1} 是正交矩阵, 我们有

$$Z_k \hat{R}_k = Y_k = A Z_{k-1} = (Z_{k-1} Z_{k-1}^T) A Z_{k-1} = Z_{k-1} A_k = (Z_{k-1} Q_k) R_k,$$

即 $Z_k \hat{R}_k$ 和 $(Z_{k-1} Q_k) R_k$ 都是 Y_k 的 QR 分解. 由 QR 分解的唯一性可知, $Z_k = Z_{k-1} Q_k$, $\hat{R}_k = R_k$. 所以

$$Z_k^T A Z_k = (Z_{k-1} Q_k)^T A (Z_{k-1} Q_k) = Q_k^T A_k Q_k = Q_k^T (Q_k R_k) Q_k = R_k Q_k = A_{k+1},$$

即 $A_{k+1} = Z_k^T A Z_k$. 由归纳法可知, 定理结论成立. □

4.3.5 QR 迭代的收敛性

定理 4.3 设 $A = V \Lambda V^{-1} \in \mathbb{R}^{n \times n}$, 其中 $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, 且 $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$. 若 V^{-1} 的所有主子矩阵都非奇异（即 V^{-1} 存在 LU 分解）, 则 A_k 的对角线以下的元素均收敛到 0.

证明. 设 $V = Q_v R_v$ 是 V 的 QR 分解, $V^{-1} = L_v U_v$ 是 V^{-1} 的 LU 分解, 其中 L_v 是单位下三角矩阵. 则

$$A^k = V \Lambda^k V^{-1} = Q_v R_v \Lambda^k L_v U_v = Q_v R_v (\Lambda^k L_v \Lambda^{-k}) \Lambda^k U_v.$$

注意到矩阵 $\Lambda^k L_v \Lambda^{-k}$ 是一个下三角矩阵, 且其 (i,j) 位置上的元素为

$$\left(\Lambda^k L_v \Lambda^{-k}\right)(i,j) = \begin{cases} 0, & i < j, \\ 1, & i = j, \\ L_v(i,j) \lambda_i^k / \lambda_j^k, & i > j. \end{cases}$$

由于当 $i > j$ 时有 $|\lambda_i / \lambda_j| < 1$, 故当 k 充分大时, $\lambda_i^k / \lambda_j^k$ 趋向于 0. 所以我们可以把 $\Lambda^k L_v \Lambda^{-k}$ 写成

$$\Lambda^k L_v \Lambda^{-k} = I + E_k,$$

其中 E_k 满足 $\lim\limits_{k \to \infty} E_k = 0$. 于是

$$A^k = Q_v R_v (I + E_k) \Lambda^k U_v = Q_v (I + R_v E_k R_v^{-1}) R_v \Lambda^k U_v. \tag{4.3}$$

对矩阵 $I + R_v E_k R_v^{-1}$ 做 QR 分解: $I + R_v E_k R_v^{-1} = Q_{E_k} R_{E_k}$. 由于 $E_k \to 0$, 所以 $Q_{E_k} \to I$, $R_{E_k} \to I$. 将其代入 (4.3) 可得

$$A^k = Q_v Q_{E_k} R_{E_k} R_v \Lambda^k U_v = Q_v Q_{E_k} D_k (D_k^{-1} R_{E_k} R_v \Lambda^k U_v), \tag{4.4}$$

其中 D_k 是对角矩阵, 其对角线元素的模均为 1, 它使得上三角矩阵 $D_k^{-1} R_{E_k} R_v \Lambda^k U_v$ 的对角线元素均为正. 这样, (4.4) 就构成 A^k 的 QR 分解. 又由 (4.2) 可知 $A^k = \tilde{Q}_k \tilde{R}_k$, 根据 QR 分解的唯一性, 我们可得

$$\tilde{Q}_k = Q_v Q_{E_k} D_k, \qquad \tilde{R}_k = D_k^{-1} R_{E_k} R_v \Lambda^k U_v.$$

所以由 (4.1) 可知

$$\begin{aligned} A_{k+1} &= \tilde{Q}_k^T A \tilde{Q}_k \\ &= (Q_v Q_{E_k} D_k)^T V \Lambda V^{-1} Q_v Q_{E_k} D_k \\ &= D_k^T Q_{E_k}^T Q_v^T Q_v R_v \Lambda R_v^{-1} Q_v^{-1} Q_v Q_{E_k} D_k \\ &= D_k^T Q_{E_k}^T R_v \Lambda R_v^{-1} Q_{E_k} D_k. \end{aligned}$$

由于 $Q_{E_k} \to I$, 所以当 $k \to \infty$ 时, A_{k+1} 收敛到一个上三角矩阵. 收敛速度取决于

$$\max_{1 \leq i < n} \left| \frac{\lambda_{i+1}}{\lambda_i} \right|$$

的大小. □

需要指出的是, 由于 D_k 的元素不一定收敛, 故 A_{k+1} 对角线以上（不含对角线）的元素不一定收敛, 但这不妨碍 A_{k+1} 的对角线元素收敛到 A 的特征值（即 A_{k+1} 的对角线元素是收敛的）.

4.3.6 带位移的 QR 迭代

为了加快 QR 迭代的收敛速度, 我们可以采用位移策略和反迭代思想.

算法 4.6 带位移的 QR 迭代算法 (QR Iteration with shift)

1: Set $A_1 = A$ and $k = 1$
2: **while** not convergence **do**
3:     Choose a shift σ_k
4:     $A_k - \sigma_k I = Q_k R_k$ % QR 分解
5:     Compute $A_{k+1} = R_k Q_k + \sigma_k I$
6:     $k = k + 1$
7: **end while**

与不带位移的 QR 迭代一样, 我们有

$$A_{k+1} = R_k Q_k + \sigma_k I = (Q_k^T Q_k) R_k Q_k + \sigma_k I = Q_k^T (A_k - \sigma_k I) Q_k + \sigma_k I = Q_k^T A_k Q_k$$

所以, 带位移的 QR 算法中所得到的矩阵 A_k 仍然与 $A_1 = A$ 正交相似.

在带位移的 QR 迭代算法中, 一个很重要的问题就是位移 σ_k 的选取. 在前面的分析中我们已经知道, $A_{k+1}(n, n)$ 将收敛到 A 的模最小的特征值, 且收敛速度取决于模最小特征值与模第二小特征值之间的比值. 显然, 若 σ_k 就是 A 的一个特征值, 则 $A_k - \sigma_k I$ 的模最小特征值为 0, 故 QR 算法迭代一步就收敛. 此时

$$A_{k+1} = R_k Q_k + \sigma_k I = \begin{bmatrix} A_{k+1}^{(n-1)\times(n-1)} & * \\ 0 & \sigma_k \end{bmatrix}.$$

如果需要计算 A 的其它特征值, 则可对子矩阵 $A_{k+1}^{(n-1)\times(n-1)}$ 使用带位移的 QR 迭代算法.

通常, 如果 σ_k 与 A 的某个特征值非常接近, 则收敛速度通常会很快. 由于 $A_k(n, n)$ 收敛到 A 的一个特征值, 所以在实际使用中, 一个比较直观的位移选择策略是 $\sigma_k = A_k(n, n)$. 事实上, 这样的位移选取方法通常会使得 QR 迭代有二次收敛速度.

4.4 带位移的隐式 QR 迭代

QR 迭代算法中需要考虑的另一个重要问题就是运算量: 每一步迭代都需要做一次 QR 分解和矩阵乘积, 运算量为 $O(n^3)$. 即使每计算一个特征值只需迭代一步, 则计算所有特征值也需要 $\mathcal{O}(n^4)$ 的运算量. 这是令人无法忍受的. 下面我们就想办法将总运算量从 $\mathcal{O}(n^4)$ 减小到 $\mathcal{O}(n^3)$.

为了实现这个目标, 我们需要利用 Hessenberg 矩阵. 具体步骤如下: 首先通过相似变化将 A 转化成一个上 Hessenberg 矩阵, 然后再对这个 Hessenberg 矩阵实施**隐式 QR 迭代**. 所谓隐式 QR 迭代, 就是在 QR 迭代中, 我们不需要进行显式的 QR 分解. 这样就可以将 QR 迭代的每一步运算量从 $\mathcal{O}(n^3)$ 降低到 $\mathcal{O}(n^2)$. 从而将总的运算量降低到 $\mathcal{O}(n^3)$.

4.4.1 上 Hessenberg 矩阵

设 $H = (h_{ij}) \in \mathbb{R}^{n\times n}$, 若当 $i > j+1$ 时, 有 $h_{ij} = 0$, 则称 H 为 **上 Hessenberg 矩阵**.

定理 4.4 设 $A \in \mathbb{R}^{n\times n}$, 则存在正交矩阵 $Q \in \mathbb{R}^{n\times n}$, 使得 QAQ^T 是上 *Hessenberg* 矩阵.

下面我们给出具体的转化过程. 这里我们主要使用 Householder 变换: 对任意向量 $x \in \mathbb{R}^n$, 总存在一个 Householder 矩阵 $H = I - \beta vv^T$ 使得 $Hx = \|x\|_2 e_1$, 其中 $e_1 = [1, 0, \ldots, 0]^T$.

我们以一个 5×5 的矩阵 A 为例.

第一步: 令 $Q_1 = \mathrm{diag}(I_{1\times 1}, H_1)$, 其中 H_1 是对应于向量 $A(2:5,1)$ 的 Householder 矩阵. 于是可得

$$Q_1 A = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix}.$$

由于用 Q_1^T 右乘 Q_1A 时, 不会改变 Q_1A 第一列元素的值, 故

$$A_1 \triangleq Q_1 A Q_1^T = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix}.$$

第二步: 令 $Q_2 = \mathrm{diag}(I_{2\times 2}, H_2)$, 其中 H_2 是对应于向量 $A_1(3:5,2)$ 的 Householder 矩阵, 则用 Q_2 左乘 A_1 时, 不会改变 A_1 的第一列元素的值. 用 Q_2^T 右乘 Q_2A_1 时, 不会改变 Q_2A_1 前两列元素的值. 因此,

$$Q_2 A_1 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{bmatrix} \quad \text{和} \quad A_2 \triangleq Q_2 A_1 Q_2^T = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{bmatrix}.$$

第三步: 令 $Q_3 = \mathrm{diag}(I_{3\times 3}, H_3)$, 其中 H_3 是对应于向量 $A_2(4:5,3)$ 的 Householder 矩阵, 则有

$$Q_3A_2 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix} \quad \text{和} \quad A_3 \triangleq Q_3A_2Q_3^T = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}.$$

这时, 我们就将 A 转化成一个上 Hessenberg 矩阵, 即 $QAQ^T = A_3$ 其中 $Q = Q_3Q_2Q_1$ 是正交矩阵, A_3 是上 Hessenberg 矩阵.

下面是将任意一个矩阵转化成上 Hessenberg 矩阵的算法.

算法 4.7 上 Hessenberg 化算法 (Upper Hessenberg Reduction)

1: Set $Q = I$
2: **for** $k = 1$ to $n-2$ **do**
3: compute Householder matrix $H_k = I - \beta_k v_k v_k^T$ with respect to $A(k+1:n,k)$
4: $A(k+1:n,k:n) = H_k \cdot A(k+1:n,k:n)$
$\qquad = A(k+1:n,k:n) - \beta_k v_k \left(v_k^T A(k+1:n,k:n)\right)$
5: $A(1:n,k+1:n) = A(1:n,k+1:n) \cdot H_k^T$
$\qquad = A(1:n,k+1:n) - \beta_k A(1:n,k+1:n) v_k v_k^T$
6: $Q(k+1:n,k:n) = H_k \cdot Q(k+1:n,k:n)$
$\qquad = Q(k+1:n,k:n) - \beta_k v_k \left(v_k^T Q(k+1:n,k:n)\right)$
7: **end for**

注: 在实际计算时, 我们不需要显式地形成 Householder 矩阵 H_k.

上述算法的运算量大约为 $\frac{14}{3}n^3 + \mathcal{O}(n^2)$. 如果不需要计算特征向量, 则正交矩阵 Q 也不用计算, 此时运算量大约为 $\frac{10}{3}n^3 + \mathcal{O}(n^2)$.

上 Hessenberg 矩阵的一个很重要的性质就是在 QR 迭代中能保持形状不变.

定理 4.5 *设 $A \in \mathbb{R}^{n\times n}$ 是非奇异上 Hessenberg 矩阵, 其 QR 分解为 $A = QR$, 则 $\tilde{A} \triangleq RQ$ 也是上 Hessenberg 矩阵.*

证明. 设 $A = QR$ 是 A 的 QR 分解, 则 $Q = AR^{-1}$. 由于 R 是一个上三角矩阵, 所以 R^{-1} 也是一个上三角矩阵. 因此 Q 的第 j 列是 A 的前 j 列的线性组合. 又 A 是上 Hessenberg 矩阵, 所以 Q 也是一个上 Hessenberg 矩阵.

相类似地, 我们很容易验证 RQ 也是一个上 Hessenberg 矩阵. 所以结论成立. □

☞ 若 A 是奇异的, 也可以通过选取适当的 Q, 使得定理 4.5 结论成立.

由这个性质可知, 如果 $A \in \mathbb{R}^{n\times n}$ 是上 Hessenberg 矩阵, 则 QR 迭代中每一步的运算量可大大降低.

Hessenberg 矩阵还有一个性质, 就是在 QR 迭代过程中能保持不可约性.

定理 4.6 设 $A \in \mathbb{R}^{n\times n}$ 是不可约上 Hessenberg 矩阵, 其 QR 分解为 $A = QR$, 则 $\tilde{A} \triangleq RQ$ 也是不可约上 Hessenberg 矩阵.

证明. 留作练习. □

推论 4.7 设 $A \in \mathbb{R}^{n\times n}$ 是不可约上 Hessenberg 矩阵, 则在带位移的 QR 迭代中, 所有的 A_k 均是不可约上 Hessenberg 矩阵.

4.4.2 隐式 QR 迭代

在 QR 迭代中, 我们需要先做 QR 分解 $A_k = Q_k R_k$, 然后再计算 $A_{k+1} = R_k Q_k$. 但事实上, 我们可以将这个过程进一步简化, 即不用计算 A_k 的 QR 分解, 可以直接计算 A_{k+1}. 这就是隐式 QR 迭代.

我们这里考虑不可约的上 Hessenberg 矩阵, 即 A 的下次对角线元素都不为 0. 事实上, 若 A 是可约的, 则 A 就是一个块上三角矩阵, 这时 A 的特征值计算问题就转化成计算两个对角块的特征值问题. 隐式 QR 迭代的理论基础就是下面的隐式 Q 定理.

定理 4.8 (Implicit Q Theorem) 设 $H = Q^T AQ \in \mathbb{R}^{n\times n}$ 是一个不可约上 Hessenberg 矩阵, 其中 $Q \in \mathbb{R}^{n\times n}$ 是正交矩阵, 则 Q 的第 2 至第 n 列均由 Q 的第一列所唯一确定 (可相差一个符号).

证明. 设 $H = Q^T AQ$ 和 $G = V^T AV$ 都是不可约上 Hessenberg 矩阵, 其中 $Q = [q_1, q_2, \ldots, q_n]$, $V = [v_1, v_2, \ldots, v_n]$ 都是正交矩阵, 且 $q_1 = v_1$. 下面我们只需证明 $q_i = v_i$ 或 $q_i = -v_i$, $i = 2, 3, \ldots, n$, 即证明

$$W = V^T Q = \mathrm{diag}(1, \pm 1, \ldots, \pm 1).$$

记 $W = [w_1, w_2, \ldots, w_n]$, $H = [h_{ij}]$. 则有

$$GW = GV^T Q = (V^T AV)V^T Q = V^T AQ = V^T Q(Q^T AQ) = V^T QH = WH,$$

即

$$Gw_i = \sum_{j=1}^{i+1} h_{ji} w_j, \quad i = 1, 2, \ldots, n-1.$$

所以

$$h_{i+1,i} w_{i+1} = Gw_i - \sum_{j=1}^{i} h_{ji} w_j.$$

因为 $q_1 = v_1$, 所以 $w_1 = [1, 0, ..., 0]^T$. 又 G 是上 Hessenberg 矩阵, 利用归纳法, 我们可以证明 w_i 的第 $i+1$ 到第 n 个元素均为 0. 所以 W 是一个上三角矩阵. 又 W 是正交矩阵, 所以 $W = \mathrm{diag}(1, \pm 1, \ldots, \pm 1)$. 由此, 定理结论成立. □

由于 Q_k 的其它列都是由 Q_k 的第一列唯一确定 (至多相差一个符号), 所以我们只要找到一个正交矩阵 $\tilde{Q}_k$ 使得其第一列与 Q_k 的第一列相等, 且 $\tilde{Q}_k^T A_k \tilde{Q}_k$ 为上 Hessenberg 矩阵, 则由隐式 Q 定理可知 $\tilde{Q}_k = WQ_k$, 其中 $W = \mathrm{diag}(1, \pm 1, \ldots, \pm 1)$. 于是

$$\tilde{Q}_k^T A_k \tilde{Q}_k = W^T Q_k^T A_k Q_k W = W^T A_{k+1} W.$$

又 $W^T A_{k+1} W$ 与 A_{k+1} 相似, 且对角线元素相等, 而其它元素也至多相差一个符号, 所以不会影响 A_{k+1} 的收敛性, 即下三角元素收敛到 0, 对角线元素收敛到 A 的特征值. 因此在 QR 迭代算法中, 我们可以用 $\tilde{Q}_k^T A_k \tilde{Q}_k$ 代替 $Q_k A_k Q_k^T$. 这就是隐式 QR 迭代的基本思想.

在实际计算中, 我们只需要利用 Givens 变换. 下面我们举一个例子, 具体说明如何利用隐式 Q 定理, 由 $A_1 = A$ 得到 A_2.

例 4.1 设 $A \in \mathbb{R}^{5\times 5}$ 是一个不可约上 Hessenberg 矩阵, 即

$$A_1 = A = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}.$$

第一步: 构造一个 Givens 变换

$$G_1^T \triangleq G(1,2,\theta_1) = \begin{bmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix},$$

其第一列 $[c_1, s_1, 0, \ldots, 0]^T$ 就是 $A_1 - \sigma_1 I$ 的第一列 $[a_{11} - \sigma_1, a_{21}, 0, \ldots, 0]^T$ 的单位化后的列向量, 这里 σ_0 是位移. 于是有

$$G_1^T A = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix} \quad 和 \quad A^{(1)} \triangleq G_1^T A G_1 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ + & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}.$$

与 A_1 相比较, $A^{(1)}$ 在 $(3,1)$ 位置上多出一个非零元, 我们把它记为 “ + ”, 并称之为 bulge. 在下面的计算过程中, 我们的目标就是将其 “赶” 出矩阵, 从而得到一个新的上 Hessenberg 矩阵, 即 A_2.

第二步: 为了消去这个 bulge, 我们可以构造 Givens 变换

$$G_2^T \triangleq G(2,3,\theta_2) = \begin{bmatrix} 1 & & & & \\ & c_2 & s_2 & & \\ & -s_2 & c_2 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \quad 使得 \quad G_2^T A^{(1)} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}.$$

为了保持与原矩阵的相似性, 需要再右乘 G_2, 所以

$$A^{(2)} \triangleq G_2^T A^{(1)} G_2 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & + & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}.$$

此时, bugle 从 $(3,1)$ 位置被 "赶" 到 $(4,2)$ 位置.

第三步: 与第二步类似, 构造 Givens 变换

$$G_3^T \triangleq G(3,4,\theta_3) = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & c_3 & s_3 & \\ & & -s_3 & c_3 & \\ & & & & 1 \end{bmatrix} \quad \text{使得} \quad G_3^T A^{(2)} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}.$$

这时

$$A^{(3)} \triangleq G_3^T A^{(2)} G_3 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & + & * & * \end{bmatrix}.$$

于是, bugle 又从 $(4,2)$ 位置又被 "赶" 到 $(5,3)$ 位置.

第四步: 再次构造 Givens 变换

$$G_4^T \triangleq G(4,5,\theta_4) = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & c_4 & s_4 \\ & & & -s_4 & c_4 \end{bmatrix} \quad \text{使得} \quad G_4^T A^{(3)} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

于是

$$A^{(4)} \triangleq G_4^T A^{(3)} G_4 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}.$$

现在, bulge 已经被 "赶" 出矩阵, 且

$$A^{(4)} = G_4^T G_3^T G_2^T G_1^T A_1 G_1 G_2 G_3 G_4 = \tilde{Q}_1^T A_1 \tilde{Q}_1,$$

其中 $\tilde{Q}_1 = G_1 G_2 G_3 G_4$. 通过直接计算可知, $\tilde{Q}_1$ 的第一列为 $[c_1, s_1, 0, 0, 0]^T$, 即 $A_1 - \sigma_1 I$ 的第一列的单位化. 根据隐式 Q 定理, $A_2 \triangleq A^{(4)} = \tilde{Q}_1^T A_1 \tilde{Q}_1$ 就是我们所需要的矩阵.

如果 $A \in \mathbb{R}^{n\times n}$ 是上 Hessenberg 矩阵, 则使用上面的算法, 带位移 QR 迭代中每一步的运算量为 $6n^2 + O(n)$.

4.4.3 位移的选取

在带位移的 QR 迭代算法中, 位移的选取非常重要. 通常, 位移越离特征值越近, 收敛速度就越快. 由习题 4.10 可知, 如果位移 σ 与某个特征值非常接近, 则 $A_k(n,n)-\sigma$ 就非常接近于 0. 这说明 $A_k(n,n)$ 通常会首先收敛到 A 的一个特征值, 所以 $\sigma = A_k(n,n)$ 是一个不错的选择. 但是, 如果这个特征值是复数, 这种位移选取方法就可能失效.

下面我们介绍一种针对共轭复特征值的位移选取方法.

假设 $\sigma \in \mathbb{C}$ 是 A 的某个复特征值 λ 的一个很好的近似, 则其共轭 $\bar{\sigma}$ 也应该是 $\bar{\lambda}$ 的一个很好的近似, 因此我们可以考虑做**双位移**, 即先以 σ 为位移, 然后再以 $\bar{\sigma}$ 为位移, 如此不断交替. 这样就有

$$\begin{aligned} A_1 - \sigma I &= Q_1R_1, \\ A_2 &= R_1Q_1 + \sigma I, \\ A_2 - \bar{\sigma} I &= Q_2R_2, \\ A_3 &= R_2Q_2 + \bar{\sigma} I. \end{aligned} \tag{4.5}$$

容易验证

$$A_3 = Q_2^T A_2 Q_2 = Q_2^* Q_1^* A_1 Q_1 Q_2 = Q^* A_1 Q,$$

其中 $Q = Q_1Q_2$. 我们注意到 σ 可能是复的, 所以 Q_1 和 Q_2 都可能是复矩阵. 但我们却可以选取适当的 Q_1 和 Q_2, 使的 $Q = Q_1Q_2$ 是实正交矩阵.

引理 4.1 *在双位移 QR 迭代 (4.5) 中, 我们可以选取酉矩阵 Q_1 和 Q_2 使得 $Q = Q_1Q_2$ 是实矩阵.*

证明. 由于

$$Q_2R_2 = A_2 - \bar{\sigma} I = R_1Q_1 + (\sigma - \bar{\sigma})I,$$

所以

$$\begin{aligned} Q_1Q_2R_2R_1 &= Q_1\big(R_1Q_1 + (\sigma-\bar{\sigma})I\big)R_1 \\ &= Q_1R_1Q_1R_1 + (\sigma-\bar{\sigma})Q_1R_1 \\ &= (A_1-\sigma I)^2 + (\sigma-\bar{\sigma})(A_1-\sigma I) \\ &= A_1^2 - (\sigma+\bar{\sigma})A_1 + \bar{\sigma}\sigma I, \\ &= A_1^2 - 2\mathrm{Re}(\sigma)A_1 + |\sigma|^2 I \\ &\in \mathbb{R}^{n\times n}. \end{aligned}$$

又 Q_1Q_2 是酉矩阵, R_2R_1 是上三角矩阵, 故 $(Q_1Q_2)(R_2R_1)$ 是实矩阵 $A_1^2 - 2\mathrm{Re}(\sigma)A_1 + |\sigma|^2 I = (A_1-\sigma I)(A_1-\bar{\sigma} I)$ 的 QR 分解. 所以 Q_1Q_2 和 R_2R_1 都可以是实矩阵. □

由这个引理可知, 存在 Q_1 和 Q_2, 使的 $Q = Q_1Q_2$ 是实正交矩阵, 从而 $A_3 = Q^TA_1Q$ 也是实矩阵. 这时, 在迭代过程 (4.5) 中, 我们无需计算 A_2, 可直接由 A_1 计算出 A_3. 具体计算还是根据隐式 Q 定理: 我们只要找到一个实正交矩阵 Q, 使得其第一列与 $A_1^2 - 2\mathrm{Re}(\sigma)A_1 + |\sigma|^2 I$ 的第一列平行, 并且 $A_3 = Q^TA_1Q$ 是上 Hessenberg 矩阵.

由于 $A_1^2 - 2\mathrm{Re}(\sigma)A_1 + |\sigma|^2 I$ 的第一列为

$$\begin{bmatrix} a_{11}^2 + a_{12}a_{21} - 2\mathrm{Re}(\sigma)a_{11} + |\sigma|^2 \\ a_{21}(a_{11} + a_{22} - 2\mathrm{Re}(\sigma)) \\ a_{21}a_{32} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{4.6}$$

所以 Q 的第一列是上述向量的单位化. 其它过程可以通过隐式 QR 迭代来实现. 但此时的 "bulge" 是一个 2×2 的小矩阵. 因此, 在双位移隐式 QR 迭代过程中, 我们需要使用 Householder 变换.

需要指出的是, 双位移 QR 迭代算法中的运算都是实数运算.

下面我们举一个例子, 具体说明如何在实数运算下实现双位移隐式 QR 迭代算法.

例 4.2 设 $A \in \mathbb{R}^{6\times 6}$ 是一个不可约上 Hessenberg 矩阵, 即

$$A_1 = A = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}.$$

第一步: 构造一个正交矩阵 $H_1 = \begin{bmatrix} \tilde{H}_1^T & 0 \\ 0 & I_{3\times 3} \end{bmatrix}$, 其中 $\tilde{H}_1 \in \mathbb{R}^{3\times 3}$, 使得其第一列与向量 (4.6) 平行. 于是有

$$H_1^T A = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ + & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix} \quad 和 \quad A^{(1)} \triangleq H_1^T A H_1 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ + & * & * & * & * & * \\ + & + & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}.$$

与 A_1 相比较, $A^{(1)}$ 在 $(3,1)$, $(4,1)$ 和 $(4,2)$ 位置上出现 bulge. 在下面的计算过程中, 我们的目标就是把它们 "赶" 出矩阵, 从而得到一个新的上 Hessenberg 矩阵, 即 A_1.

第二步: 令 $H_2 = \begin{bmatrix} I_{1\times 1} & 0 & 0 \\ 0 & \tilde{H}_2^T & 0 \\ 0 & 0 & I_{3\times 3} \end{bmatrix}$, 其中 $\tilde{H}_2 \in \mathbb{R}^{3\times 3}$ 是对应于 $A(2:4,1)$ 的 Householder

变换, 使得

$$H_2^T A^{(1)} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & + & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix} \quad 和 \quad A^{(2)} \triangleq H_2^T A^{(1)} H_2 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & + & * & * & * & * \\ 0 & + & + & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}.$$

这时, 我们将 bugle 向右下角方向 "赶" 了一个位置.

第三步: 与第二步类似, 令 $H_3 = \begin{bmatrix} I_{2\times 2} & 0 & 0 \\ 0 & \tilde{H}_3^T & 0 \\ 0 & 0 & I_{1\times 1} \end{bmatrix}$, 其中 $\tilde{H}_3 \in \mathbb{R}^{3\times 3}$ 是对应于 $A(3:5,2)$ 的 Householder 变换, 使得

$$H_3^T A^{(2)} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & + & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix} \quad 和 \quad A^{(3)} \triangleq H_3^T A^{(2)} H_3 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & + & * & * & * \\ 0 & 0 & + & + & * & * \end{bmatrix}.$$

此时, bugle 又被向右下角方向 "赶" 了一个位置.

第四步: 令 $H_4 = \begin{bmatrix} I_{3\times 3} & 0 \\ 0 & \tilde{H}_4^T \end{bmatrix}$, 其中 $\tilde{H}_4 \in \mathbb{R}^{3\times 3}$ 是对应于 $A(4:6,3)$ 的 Householder 变换, 使得

$$H_4^T A^{(3)} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & + & * & * \end{bmatrix} \quad 和 \quad A^{(4)} \triangleq H_4^T A^{(3)} H_4 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & + & * & * \end{bmatrix}.$$

第五步: 此时, 只需构造一个 Givens 变换 $G_5 = \begin{bmatrix} I_{4\times 4} & 0 \\ 0 & G(4,5,\theta)^T \end{bmatrix}$, 使得

$$G_5^T A^{(4)} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix} \quad 和 \quad A^{(5)} \triangleq G_5^T A^{(4)} G_5 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}.$$

现在, bulge 已经被全部消除, 且

$$A^{(5)} = Q^T A Q,$$

其中 $Q = H_1 H_2 H_3 H_4 G_5$. 通过直接计算可知, Q 的第一列即为 H_1 的第一列, 也就是向量 (4.6) 的单位化. 根据隐式 Q 定理, $A^{(5)} = Q^T A Q$ 就是双位移 QR 迭代 (4.5) 中的 A_3.

最后要考虑位移 σ 的具体选取问题. 在单位移 QR 迭代算法中, 如果 A 的特征值都是实数的话, 我们可以取 $\sigma_k = A_k(n,n)$. 推广到复共轭特征值上, 我们可以取 A_k 的右下角矩阵

$$\begin{bmatrix} A_k(n-1,n-1) & A_k(n-1,n) \\ A_k(n,n-1) & A_k(n,n) \end{bmatrix}$$

的复共轭特征值作为双位移. 这样选取的位移就是 Francis **位移**.

一般来说, 采用 Francis 位移的 QR 迭代会使得迭代矩阵的右下角收敛到一个上三角矩阵 (两个实特征值) 或一个 2×2 的矩阵 (一对共轭特征值), 而且通常会有二次渐进收敛性. 在实际计算中, 计算一个特征值一般平均只需迭代两步.

注: 判断 QR 迭代算法是否收敛主要是看 $A_k(n-1,n-2)$ (或 $A_k(n,n-1)$) 是否趋向于 0.

但需要指出的是, QR 迭代并不是对所有矩阵都收敛. 例如:

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

对于上面的矩阵, 采用 Francis 位移的 QR 迭代算法无效.

另外, 也可以考虑多重位移策略, 参见 [41].

4.4.4 收缩 Deflation

收缩 (deflation) 是指在隐式 QR 迭代过程中, 当矩阵 A_{k+1} 的某个下次对角线元素 $a_{i+1,i}$ 收敛到 0 (或者很小) 时,

To be continued ...

4.5 特征向量的计算

设 A 的特征值都是实的, $R = Q^T AQ$ 是其 Schur 标准型. 若 $Ax = \lambda x$, 则 $Ry = \lambda y$, 其中 $y = Q^T x$ 或 $x = Qy$. 故只需计算 R 对应于 λ 的特征向量 y 即可.

因为 R 的对角线元素即为 A 的特征值, 所以可设 $\lambda = R(i,i)$. 假定 λ 是单重特征值, 则方程 $(R-\lambda I)y = 0$ 即为

$$\begin{bmatrix} R_{11}-\lambda I & R_{12} & R_{13} \\ 0 & 0 & R_{23} \\ 0 & 0 & R_{33}-\lambda I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = 0,$$

即

$$(R_{11}-\lambda I)y_1 + R_{12}y_2 + R_{13}y_3 = 0, \tag{4.7}$$

$$R_{23}y_3 = 0, \tag{4.8}$$

$$(R_{33}-\lambda I)y_3 = 0, \tag{4.9}$$

其中 $R_{11} \in \mathbb{R}^{(i-1)\times(i-1)}$, $R_{33} \in \mathbb{R}^{(n-i)\times(n-i)}$. 由于 λ 是单重特征值, 故 $R_{33}-\lambda I$ 非奇异, 因此 $y_3 = 0$. 令 $y_2 = 1$, 则可得

$$y_1 = (R_{11}-\lambda I)^{-1} R_{12}.$$

因此计算特征向量 y 只需求解一个上三角线性方程组.

若 λ 是多重特征值, 则计算方法类似. 但如果 A 有复特征值, 则需要利用实 Schur 标准型, 计算较复杂.

4.6 广义特征值问题

设 $A, B \in \mathbb{R}^{n\times n}$, 若存在 $\lambda \in \mathbb{C}$ 和非零向量 $x \in \mathbb{C}^n$ 使得

$$Ax = \lambda Bx, \tag{4.10}$$

则称 λ 为矩阵对 (A, B) 的特征值, x 为相应的特征向量. 计算矩阵对 (A, B) 的特征值和特征向量就是 **广义特征值问题**. 当 $B = I$ 时, 广义特征值问题就退化为标准特征值问题. 当 B 非奇异时, 广义特征值问题就等价于标准特征值问题

$$B^{-1}Ax = \lambda x \quad 或 \quad AB^{-1}y = \lambda y,$$

其中 $y = Bx$.

容易看出, λ 是 (A, B) 的一个特征值当且仅当

$$\det(A - \lambda B) = 0. \tag{4.11}$$

若 (4.11) 对所有 $\lambda \in \mathbb{C}$ 都成立, 则称矩阵对 (A, B) 是 **奇异矩阵对**, 否则称为 **正则矩阵对**.

当 B 非奇异时, 特征方程 (4.11) 是一个 n 次多项式, 因此恰好有 n 个特征值. 当 B 奇异时, 特征方程 (4.11) 的次数低于 n, 因此方程的解的个数小于 n. 但是, 注意到 $\lambda \neq 0$ 是 (A, B) 的特征值当且仅当 $\mu = \frac{1}{\lambda}$ 是 (B, A) 的特征值. 因此, 当 B 奇异时, $\mu = 0$ 是 (B, A) 的特征值, 于是我们自然地把 $\lambda = \frac{1}{\mu} = \infty$ 当作是 (A, B) 的特征值. 所以, 广义特征值不是分布在 $\mathbb{C}$ 上, 而是分布在 $\mathbb{C} \cup \{\infty\}$ 上.

容易验证, 若 U, V 非奇异, 则矩阵对 (U^*AV, U^*BV) 的特征值与 (A, B) 是一样的. 因此我们称这种变换为 **矩阵对的等价变换**. 如果 U, V 是酉矩阵, 则称为 **酉等价变换**.

4.6.1 广义 Schur 分解

广义 Schur 分解 是矩阵对在酉等价变化下的最简形式.

定理 4.9 (广义 Schur 分解) 设 $A, B \in \mathbb{C}^{n\times n}$, 则存在酉矩阵 $Q, Z \in \mathbb{C}^{n\times n}$, 使得

$$Q^*AZ = R_A, \quad Q^*BZ = R_B, \tag{4.12}$$

其中 $R_A, R_B \in \mathbb{C}^{n\times n}$ 都是上三角矩阵. 此时矩阵对 (A, B) 的特征值为 R_A 和 R_B 的对角线元素的比值, 即

$$\lambda_i = \frac{R_A(i,i)}{R_B(i,i)}, \quad i = 1, 2, \ldots, n.$$

当 $R_B(i,i) = 0$ 时, 对应的特征值 $\lambda_i = \infty$.

证明. 参见 [53]. □

与实 Schur 分解类似, 当 A, B 都是实矩阵时, 我们有相应的 **广义实 Schur 分解**.

定理 4.10 (广义实 Schur 分解) 设 $A, B \in \mathbb{R}^{n\times n}$, 则存在正交矩阵 $Q, Z \in \mathbb{R}^{n\times n}$, 使得

$$Q^TAZ = T_A, \quad Q^TBZ = T_B, \tag{4.13}$$

其中 $T_A, T_B \in \mathbb{R}^{n\times n}$ 都是拟上三角矩阵.

证明. 参见 [53]. □

4.6.2 QZ 迭代

QZ 迭代是用于计算 (A, B) 的广义 Schur 分解的算法, 是 QR 算法的自然推广, 实质上可以看作是将 QR 算法作用到矩阵 AB^{-1} 上.

详细算法可参见 [25, 53].

4.7 课后习题

4.1 设 $J = \begin{bmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix} \in \mathbb{C}^{m\times m}$, 计算其特征值 λ 对应的特征向量和左特征向量.

4.2 设 $A \in \mathbb{C}^{n\times n}$. 证明:

(1) 若 A 是上三角矩阵, 且 $A^*A = AA^*$, 则 A 必定是对角矩阵;

(2) A 是正规矩阵的充要条件是 A 酉相似于一个对角矩阵;

(3) 设 $\lambda_1, \lambda_2, \ldots, \lambda_n$ 是 A 的特征值, 则 A 是正规矩阵的充要条件是 $\sum_{i=1}^{n} |\lambda_i|^2 = \|A\|_F^2$.

4.3 设 $\lambda_1, \lambda_2 \in \mathbb{C}$ 是 $A \in \mathbb{C}^{n\times n}$ 的两个互不相同的特征值, $x \in \mathbb{C}^n$ 是 λ_1 的特征向量, $y \in \mathbb{C}^n$ 是 λ_2 的左特征向量. 证明: $y^*x = 0$.

4.4 设 $A \in \mathbb{C}^{m\times n}, B \in \mathbb{C}^{n\times m}$, 证明: AB 与 BA 具有相同的非零特征值.

4.5 设 $A \in \mathbb{C}^{m\times m}, B \in \mathbb{C}^{n\times n}, C \in \mathbb{C}^{m\times n}$, 考虑矩阵方程

$$AX - XB = C.$$

(1) 证明: 当 A 和 B 没有共同特征值时, 矩阵方程存在唯一解.
(提示: 利用 Kronecker 积, 上述矩阵方程等价于 $(I_n \otimes A + B^T \otimes I_m)\mathrm{vec}(X) = \mathrm{vec}(C)$, 其中 $\mathrm{vec}(\cdot)$ 表示将矩阵按列排列得到的向量)

(2) 当 A 和 B 没有共同特征值时, 给出求解算法.
(提示: 利用 Schur 分解, 将 A 和 B 转化为相应的上三角矩阵)

4.6 设 $A \in \mathbb{C}^{n\times n}$ 可对角化, 其特征值为 $\lambda_1, \lambda_2, \ldots, \lambda_n$.
证明: $\sum_{i=1}^{n} |\lambda_i|^2 = \min_{\det(S)\neq 0} \|S^{-1}AS\|_F^2$.

4.7 设

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ & A_{22} & \cdots & A_{2k} \\ & & \ddots & \vdots \\ & & & A_{kk} \end{bmatrix},$$

其中 A_{ii} 都是方阵. 证明: A 的特征值即为对角块 $A_{11}, A_{22}, \ldots, A_{kk}$ 的特征值的并.

4.8 设 $A = QH = VG$ 是非奇异矩阵 $A \in \mathbb{C}^{n\times n}$ 的两个 QR 分解, 其中 $Q, V \in \mathbb{C}^{n\times n}$ 是酉矩阵, $H, G \in \mathbb{C}^{n\times n}$ 是上三角矩阵. 证明: 若存在 $\alpha_1 \in \mathbb{C}$ 满足 $|\alpha_1| = 1$, 使得 $q_1 = \alpha_1 v_1$, 则存在对角矩阵 $W = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{C}^{n\times n}$ 满足 $|\alpha_i| = 1$, 使得

$$Q = VW, \quad H = W^{-1}G.$$

4.9 设 $H \in \mathbb{R}^{n\times n}$ 是不可约上 Hessenberg 矩阵, 证明: 存在对角矩阵 D 使得 $D^{-1}HD$ 的次对角元素都为 1. 若 H 的次对角元素都小于 1 或都大于 1, 估计 D 的谱条件数 $\kappa_2(D)$.

4.10 设 $H = [h_{ij}] \in \mathbb{R}^{n\times n}$ 是上 Hessenberg 矩阵, 其 QR 分解为 $H = QR$, 其中 $R = [r_{ij}] \in \mathbb{R}^{n\times n}$ 是上三角矩阵. 证明:

$$r_{kk} > |h_{k+1,k}|, \quad k = 1, 2, \dots, n-1.$$

因此, (1) 若 H 不可约, 则 $r_{kk} > 0$, $k = 1, 2, \dots, n-1$; (2) 若 H 不可约且奇异, 则 $r_{nn} = 0$. (提示: 观察 H 的 QR 分解过程, 借助 Givens 变换)

4.11 证明定理 4.6 :
设 $A \in \mathbb{R}^{n\times n}$ 是不可约上 Hessenberg 矩阵, 其 QR 分解为 $A = QR$, 则 $\tilde{A} \triangleq RQ$ 也是不可约上 Hessenberg 矩阵.

计算题

4.12 用 Householder 变换, 将一个给定的矩阵 $A \in \mathbb{R}^{4\times 4}$ 化为上 Hessenberg 型.

4.13 考虑用 Householder 变换将矩阵上 Hessenberg 化的算法, 给出具体的乘法运算次数和加减运算次数.

4.14 证明矩阵

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & -a_{n-2} \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{bmatrix}$$

的特征多项式是

$$p(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0.$$

并利用这个结果给出计算一个多项式所有零点的实用算法.

实践题

4.15 用 Matlab 实现上 Hessenberg 化算法 4.7.
$[H, Q]$=**hessenberg**(A)

第五章　对称特征值问题

本章主要讨论对称特征值问题的计算.

5.1 对称特征值的算法综述

关于对称特征值问题的常用算法有 (直接法):

- Jacobi 迭代: 最古老的方法, 收敛速度较慢, 但精度较高, 且很适合并行计算.
- Rayleigh 商迭代: 利用 Rayleigh 商作为位移的反迭代算法, 一般具有三次收敛性.
- 对称 QR 迭代: 计算对称矩阵的特征值和特征向量的 QR 算法. 如果只需计算对称三对角矩阵的所有特征值, 则该算法是目前最快的算法 (运算量为 $O(n^2)$). 如果需要计算所有的特征值和特征向量, 则运算量约为 $6n^3$, 当 $n \leq 25$ 时, 它是首选方法.
- 分而治之法 (Divide-and-Conquer) : 计算对称三对角矩阵的特征值和特征值向量的一种快速算法. 基本思想是将大矩阵分解成小矩阵, 然后利用递推思想求特征值. 在最坏的情形下, 运算量为 $O(n^3)$, 但在实际应用中, 平均为 $O(n^{2.3})$. 如果使用快速多极子算法 (FMM) 后, 理论上的运算量可降低到 $O(n \log^p n)$, 其中 p 是一个较小的整数, 这使得分而治之算法成为目前求解 $n > 25$ 的对称三对角矩阵的特征值和特征向量的最快的方法.
- 对分法和反迭代: 对分法主要用于求解对称三对角矩阵在某个区间中的特征值, 运算量约为 $O(kn)$, 其中 k 为所需计算的特征值的个数. 反迭代用于计算特征向量, 在最佳情况下, 即特征值 “适当分离” 时, 运算量约为 $O(kn)$, 但在最差情况下, 即特征值成串地紧靠在一起时, 运算量约为 $O(k^2n)$, 而且不能保证特征向量的精度 (虽然实际上它几乎是精确的).

☞ 除了 Jacobi 迭代和 Rayleigh 商迭代外, 其余算法都需要先将对称矩阵三对角化, 这个过程大约需花费 $\frac{4}{3}n^3$ 的工作量, 如果需要计算特征向量的话, 则运算量约为 $\frac{8}{3}n^3$.

5.2 Jacobi 迭代

该算法的基本思想是通过一系列的 Jacobi 旋转 J_k 将 A 正交相似于一个对角矩阵, 即

$$A^{(0)} = A, \quad A^{(k+1)} = J_k^T A^{(k)} J_k, \quad k = 0, 1, \ldots,$$

且 $A^{(k)}$ 收敛到一个对角矩阵, 其中 J_k 为正交矩阵.

Jacobi 旋转 J_k 的构造

我们通常选取 J_k 为 Givens 变换, 即

$$J_k = G(i_k, j_k, \theta_k) = \begin{array}{c} \\ \\ \\ i_k \\ \\ j_k \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccccc} & & & i_k & & j_k & & & \end{array} \\ \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \cos\theta_k & & -\sin\theta_k & & & \\ & & & & \ddots & & & & \\ & & & \sin\theta_k & & \cos\theta_k & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix} \end{array}.$$

由于 $A^{(k)}$ 是对称矩阵, 所以可以选取适当的 θ_k, 将 $A^{(k)}(i,j)$ 和 $A^{(k)}(j,i)$ 化为 0.

引理 5.1 设 $A \in \mathbb{R}^{2\times 2}$ 是对称矩阵, 则存在 *Givens* 变换 $G \in \mathbb{R}^{2\times 2}$ 使得 G^TAG 为对角阵.

证明. 设

$$A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \quad G = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

则

$$\begin{aligned} G^TAG &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^T \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \\ &= \begin{bmatrix} a\cos^2\theta + c\sin^2\theta + b\sin 2\theta & \frac{1}{2}(c-a)\sin 2\theta + b\cos 2\theta \\ \frac{1}{2}(c-a)\sin 2\theta + b\cos 2\theta & a\sin^2\theta + c\cos^2\theta - b\sin 2\theta \end{bmatrix} \end{aligned}$$

令 $\frac{1}{2}(c-a)\sin 2\theta + b\cos 2\theta = 0$ 即得

$$\frac{a-c}{2b} = \cot 2\theta = \frac{1-\tan^2\theta}{2\tan\theta}.$$

解得

$$\tan\theta = \frac{\operatorname{sign}(\tau)}{|\tau| + \sqrt{1+\tau^2}}, \qquad \tau = \frac{a-c}{2b}.$$

故引理结论成立. □

为了使得 $A^{(k)}$ 收敛到一个对角矩阵, 其非对角元素必须趋向于 0. 记 off(A) 为所有非对角元素的平方和, 即

$$\mathrm{off}(A) = \sum_{i \neq j} a_{ij}^2 = \|A\|_F^2 - \sum_{i=1}^{n} a_{ii}^2\,,$$

则我们的目标就是使得 off(A) 尽快趋向于 0.

引理 5.2 设 $A = [a_{ij}]_{n\times n} \in \mathbb{R}^{n\times n}$ 是对称矩阵, $\hat{A} = J^T A J = [\hat{a}_{ij}]_{n\times n}$, $J = G(i, j, \theta)$, 其中 θ 的选取使得 $\hat{a}_{ij} = \hat{a}_{ji} = 0$, 则

$$\mathrm{off}(\hat{A}) = \mathrm{off}(A) - 2a_{ij}^2.$$

证明. 设 $A = [a_1, a_2, \ldots, a_n]$. 令 $\tilde{A} = J^T A = [\tilde{a}_{ij}]_{n\times n}$. 由于 J 是正交阵, 故

$$\|J^T a_k\|_2 = \|a_k\|_2, \quad k = 1, 2, \ldots, n.$$

又 J^T 左乘 a_k 时, 只影响其第 i, j 个元素的值, 故由 $\|J^T a_i\|_2 = \|a_i\|_2$ 和 $\|J^T a_j\|_2 = \|a_j\|_2$ 可得

$$\tilde{a}_{ii}^2 + \tilde{a}_{ji}^2 = a_{ii}^2 + a_{ji}^2, \quad \tilde{a}_{ij}^2 + \tilde{a}_{jj}^2 = a_{ij}^2 + a_{jj}^2. \tag{5.1}$$

同理, 由 $\hat{A} = \tilde{A}J$ 可得

$$\hat{a}_{ii}^2 + \hat{a}_{ij}^2 = \tilde{a}_{ii}^2 + \tilde{a}_{ij}^2, \quad \hat{a}_{ji}^2 + \hat{a}_{jj}^2 = \tilde{a}_{ji}^2 + \tilde{a}_{jj}^2. \tag{5.2}$$

又 $\hat{a}_{ij} = \hat{a}_{ji} = 0$, 故

$$\hat{a}_{ii}^2 + \hat{a}_{jj}^2 = a_{ii}^2 + a_{jj}^2 + a_{ij}^2 + a_{ji}^2 = a_{ii}^2 + a_{jj}^2 + 2a_{ij}^2.$$

由于 $J^T A J$ 只影响 A 的第 i, j 行和第 i, j 列, 故对角线元素中只有 a_{ii} 和 a_{jj} 受影响. 所以

$$\sum_{k=1}^{n} \hat{a}_{kk}^2 = \sum_{k=1}^{n} a_{kk}^2 + 2a_{ij}^2,$$

故

$$\mathrm{off}(\hat{A}) = \|\hat{A}\|_2^2 - \sum_{k=1}^{n} \hat{a}_{kk}^2 = \|A\|_2^2 - \sum_{k=1}^{n} a_{kk}^2 - 2a_{ij}^2 = \mathrm{off}(A) - 2a_{ij}^2,$$

即引理结论成立. □

由此可知, off$(A^{(k)})$ 总是不断减小的. 下面给出 Jacobi 迭代算法.

算法 5.1 Jacobi 迭代算法

1: Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$
2: **if** eigenvectors are desired **then**
3:     set $J = I$ and $shift = 1$
4: **end if**
5: **while** not converge **do**
6:     choose an index pair (i, j) such that $a_{ij} \neq 0$

7: $\tau = (a_{ii} - a_{jj})/(2a_{ij})$
8: $t = \text{sign}(\tau)/(|\tau| + \sqrt{1+\tau^2})$ % 计算 $\tan\theta$
9: $c = 1/\sqrt{1+t^2}$ % 计算 $\cos\theta$
10: $s = c \cdot t$ % 计算 $\sin\theta$
11: $A = G(i,j,\theta)^T A G(i,j,\theta)$ % 实际计算时不需要做矩阵乘积
12: **if** $shift = 1$ **then**
13: $J = J \cdot G(i,j,\theta)$
14: **end if**
15: **end while**

该算法涉及到 a_{ij} 的选取问题, 一种直观的选取方法就是使得 a_{ij} 为所有非对角元素中绝对值最大的一个, 于是我们就得到下面的经典 Jacobi 算法.

算法 5.2 经典 Jacobi 迭代算法

1: Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$
2: **if** eigenvectors are desired **then**
3: set $J = I$ and $shift = 1$
4: **end if**
5: **while** off$(A) > tol$ **do**
6: choose (i,j) such that $|a_{ij}| = \max_{k\neq l} |a_{kl}|$
7: $\tau = (a_{ii} - a_{jj})/(2a_{ij})$
8: $t = \text{sign}(\tau)/(|\tau| + \sqrt{1+\tau^2})$
9: $c = 1/\sqrt{1+t^2}$
10: $s = c \cdot t$
11: $A = G(i,j,\theta)^T A G(i,j,\theta)$
12: **if** $shift = 1$ **then**
13: $J = J \cdot G(i,j,\theta)$
14: **end if**
15: **end while**

可以证明, 经典 Jacobi 算法至少是线性收敛的.

定理 5.1 对于经典 *Jacobi* 算法 5.2, 有

$$\text{off}(A^{(k+1)}) \leq \left(1 - \frac{1}{N}\right) \text{off}(A^{(k)}), \quad N = \frac{n(n-1)}{2}.$$

故 k 步迭代后, 有

$$\text{off}(A^{(k)}) \leq \left(1 - \frac{1}{N}\right)^k \text{off}(A^{(0)}) = \left(1 - \frac{1}{N}\right)^k \text{off}(A).$$

证明. 由于在经典 Jacobi 算法 5.2 中, $|a_{ij}| = \max_{k\neq l} |a_{kl}|$, 故 $\text{off}(A^{(k)}) \leq n(n+1)\left(a_{ij}^{(k)}\right)^2$, 即

$$2\left(a_{ij}^{(k)}\right)^2 \geq \frac{1}{N}\text{off}(A^{(k)}), \quad N = \frac{n(n-1)}{2}.$$

所以由引理 5.2 可知

$$\text{off}(A^{(k+1)}) = \text{off}(A^{(k)}) - \left(a_{ij}^{(k)}\right)^2 \leq \left(1 - \frac{1}{N}\right)\text{off}(A^{(k)}).$$

□

事实上, 经典 Jacobi 算法最终是二次局部收敛的.

定理 5.2 *经典 Jacobi 算法 5.2 是 N 步局部二次收敛的, 即对足够大的 k, 有*

$$\text{off}\left(A^{(k+N)}\right) = O\left(\text{off}^2\left(A^{(k)}\right)\right).$$

由于在经典 Jacobi 算法中, 每一步都要寻找绝对值最大的非对角元, 比较费时, 因此它并不实用. 我们可以通过逐行扫描来选取 (i,j), 这就是循环 Jacobi 迭代算法.

算法 5.3 循环 Jacobi 迭代算法 (逐行扫描)

1: Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$
2: **if** eigenvectors are desired **then**
3:     set $J = I$ and $shift = 1$
4: **end if**
5: **while** off(A)> tol **do**
6:     **for** $i = 1$ to $n-1$ **do**
7:         **for** $j = i+1$ to n **do**
8:             **if** $a_{ij} \neq 0$ **then**
9:                 $\tau = (a_{ii} - a_{jj})/(2a_{ij})$
10:                 $t = \text{sign}(\tau)/(|\tau| + \sqrt{1+\tau^2})$
11:                 $c = 1/\sqrt{1+t^2}$
12:                 $s = c \cdot t$
13:                 $A = G(i,j,\theta)^T A G(i,j,\theta)$
14:                 **if** $shift = 1$ **then**
15:                     $J = J \cdot G(i,j,\theta)$
16:                 **end if**
17:             **end if**
18:         **end for**
19:     **end for**
20: **end while**

循环 Jacobi 也具有局部二次收敛性 [43, page 270].

5.3 Rayleigh 商迭代

在反迭代算法中, x_i 的 Rayleigh 商是特征值 λ_i 的近似, 故我们把它作为位移, 于是就得到下面的 Rayleigh 商迭代算法.

算法 5.4 Rayleigh 商迭代

1: Given an initial guess x_0 with $\|x_0\|_2 = 1$
2: compute the Rayleigh quotient $\rho_0 = x_0^T A x_0$
3: set $i = 1$
4: **while** not converge **do**
5: $\quad \sigma = \rho_{i-1}$
6: $\quad y_i = (A - \sigma I)^{-1} x_{i-1}$
7: $\quad x_i = y_i / \|y_i\|_2$
8: $\quad \rho_i = x_i^T A x_i$
9: $\quad i = i + 1$
10: **end while**

关于 Rayleigh 商迭代的收敛性, 我们有下面的结论.

定理 5.3 如果特征值是单重的, 则当误差足够小时, *Rayleigh* 商迭代中每步迭代所得的正确数字的位数增至三倍, 即 *Rayleigh* 商迭代是局部三次收敛的.

证明. 设 $A = Q\Lambda Q^T$, 令 $\hat{x}_i = Q^T x_i$, 则在 Rayleigh 商迭代算法中

$$\rho_i = x_i^T A x_i = \hat{x}_i^T Q^T A Q \hat{x}_i = \hat{x}_i^T \Lambda \hat{x}_i.$$

令 $\hat{y}_i = Q^T y_i$, 则

$$\hat{y}_i = Q^T (A - \rho_{i-1} I)^{-1} x_i = (Q^T A Q - \rho_{i-1} I)^{-1} \hat{x}_{i-1} = (\Lambda - \rho_{i-1} I)^{-1} \hat{x}_{i-1},$$

即, "以初始向量 x_0 对 A 做 Rayleigh 商迭代" 等价于 "以初始向量 $\hat{x}_0$ 对 Λ 做 Rayleigh 商迭代", 即它们有相同的收敛性. 因此, 不失一般性, 我们可以假定 $A = \Lambda$ 为对角阵, 此时 A 的特征向量为 e_i, $i = 1, 2, \ldots, n$.

我们假定 x_i 收敛到 e_1. 令 $d_i = x_i - e_1$, 则 $\|d_i\|_2 \to 0$. 为了证明算法具有局部三次收敛, 我们需要证明: 当 $\varepsilon_i = \|d_i\|_2$ 充分小时, 有 $\varepsilon_{i+1} = \|d_{i+1}\|_2 = \|x_{i+1} - e_1\|_2 = O(\varepsilon_i^3)$.

我们注意到

$$1 = x_i^T x_i = (e_1 + d_i)^T (e_1 + d_i) = 1 + 2d_i(1) + d_i^T d_i = 1 + 2d_i(1) + \varepsilon_i^2,$$

其中 $d_i(1)$ 表示 d_i 的第一个元素. 故 $d_i(1) = -\varepsilon_i^2/2$. 所以

$$\rho_i = x_i^T \Lambda x_i = (e_1 + d_i)^T \Lambda (e_1 + d_i) = e_i^T \Lambda e_1 + 2e_1^T \Lambda d_i + d_i^T \Lambda d_i \triangleq \lambda_1 - \eta,$$

其中 $\eta = -(2e_1^T \Lambda d_i + d_i^T \Lambda d_i) = -2\lambda_1 d_i(1) - d_i^T \Lambda d_i = \lambda_1 \varepsilon_i^2 - d_i^T \Lambda d_i$. 于是

$$|\eta| \leq |\lambda_1| \varepsilon_i^2 + \|\Lambda\|_2 \cdot \|d_i\|_2^2 \leq 2\|\Lambda\|_2 \, \varepsilon_i^2.$$

由 Rayleigh 商算法 5.4 可知

$$\begin{aligned}
y_{i+1} &= (\Lambda - \rho_i I)^{-1} x_i \\
&= \left[\frac{x_i(1)}{\lambda_1 - \rho_i}, \frac{x_i(2)}{\lambda_2 - \rho_i}, \dots, \frac{x_i(n)}{\lambda_n - \rho_i}\right]^T \\
&= \left[\frac{1 + d_i(1)}{\lambda_1 - \rho_i}, \frac{d_i(2)}{\lambda_2 - \rho_i}, \dots, \frac{d_i(n)}{\lambda_n - \rho_i}\right]^T \\
&= \left[\frac{1 - \varepsilon_i^2/2}{\eta}, \frac{d_i(2)}{\lambda_2 - \lambda_1 + \eta}, \dots, \frac{d_i(n)}{\lambda_n - \lambda_1 + \eta}\right]^T \\
&= \frac{1 - \varepsilon_i^2/2}{\eta}\left[1, \frac{d_i(2)\eta}{(1 - \varepsilon_i^2/2)(\lambda_2 - \lambda_1 + \eta)}, \dots, \frac{d_i(n)\eta}{(1 - \varepsilon_i^2/2)(\lambda_n - \lambda_1 + \eta)}\right]^T \\
&\triangleq \frac{1 - \varepsilon_i^2/2}{\eta} \cdot (e_1 + \hat{d}_{i+1}).
\end{aligned}$$

其中

$$\hat{d}_{i+1} = \left[0, \frac{d_i(2)\eta}{(1 - \varepsilon_i^2/2)(\lambda_2 - \lambda_1 + \eta)}, \dots, \frac{d_i(n)\eta}{(1 - \varepsilon_i^2/2)(\lambda_n - \lambda_1 + \eta)}\right]^T.$$

因为 λ_1 是单重特征值, 所以

$$\mathrm{gap}(\lambda_1, \Lambda) \triangleq \min_{i \neq 1} |\lambda_i - \lambda_1| > 0,$$

故当 ε_i 足够小时,

$$|\lambda_j - \lambda_1 + \eta| \geq |\lambda_j - \lambda_1| - |\eta| \geq \mathrm{gap}(\lambda_1, \Lambda) - |\eta| \geq \mathrm{gap}(\lambda_1, \Lambda) - 2\|\Lambda\|_2\, \varepsilon_i^2 > 0.$$

于是我们有

$$\left\|\hat{d}_{i+1}\right\|_2 \leq \frac{\|d_i\|_2\, |\eta|}{(1 - \varepsilon_i^2/2)(\mathrm{gap}(\lambda_1, \Lambda) - |\eta|)} \leq \frac{2\|\Lambda\|_2\, \varepsilon_i^3}{(1 - \varepsilon_i^2/2)(\mathrm{gap}(\lambda_1, \Lambda) - |\eta|)},$$

即 $\left\|\hat{d}_{i+1}\right\|_2 = O(\varepsilon_i^3)$. 又

$$1 - \left\|\hat{d}_{i+1}\right\|_2 \leq \left\|e_1 + \hat{d}_{i+1}\right\|_2 \leq 1 + \left\|\hat{d}_{i+1}\right\|_2,$$

即

$$\left|1 - \left\|e_1 + \hat{d}_{i+1}\right\|_2\right| \leq \left\|\hat{d}_{i+1}\right\|_2.$$

由于

$$x_{i+1} = \frac{y_{i+1}}{\|y_{i+1}\|_2} = \frac{e_1 + \hat{d}_{i+1}}{\left\|e_1 + \hat{d}_{i+1}\right\|_2},$$

所以

$$\begin{aligned}
\|d_{i+1}\|_2 = \|x_{i+1} - e_1\|_2 &= \frac{\left\|\left(1 - \left\|e_1 + \hat{d}_{i+1}\right\|_2\right) e_1 + \hat{d}_{i+1}\right\|_2}{\left\|e_1 + \hat{d}_{i+1}\right\|_2} \\
&\leq \frac{\left|1 - \left\|e_1 + \hat{d}_{i+1}\right\|_2\right| + \left\|\hat{d}_{i+1}\right\|_2}{\left\|e_1 + \hat{d}_{i+1}\right\|_2}
\end{aligned}$$

$$\leq \frac{2\left\|\hat{d}_{i+1}\right\|_2}{\left\|e_1 + \hat{d}_{i+1}\right\|_2}.$$

又 $\left\|\hat{d}_{i+1}\right\|_2 = O(\varepsilon_i^3)$, 故 $\varepsilon_{i+1} = \|d_{i+1}\|_2 = O(\varepsilon_i^3)$. □

5.4 对称 QR 迭代

将带位移的隐式 QR 算法运用到对称矩阵, 就得到对称 QR 迭代算法.

基本步骤

1. 对称三对角化: 利用 Householder 变换, 将 A 化为对称三对角矩阵, 即计算正交矩阵 Q 使得 $T = QAQ^T$ 为对称三对角矩阵;
2. 使用带 (单) 位移的隐式 QR 迭代算法计算 T 的特征值与特征值向量;
3. 计算 A 的特征向量.

对称三对角化

任何一个对称矩阵 $A \in \mathbb{R}^{n\times n}$ 都可以通过正交变换转化成一个对称三对角矩阵 T. 这个过程可以通过 Householder 变换来实现, 也可以通过 Givens 变换来实现.

如果 A 是一个稠密矩阵, 则 Givens 变换的总运算量大约是 Householder 变换的两倍, 因此建议使用 Householder 变换.

如果 A 是带状矩阵, 假设带宽为 $2N+1$ (即上带宽和下带宽都是 N), 且 $N \ll n$. 通过 Givens 变换进行对称三对角化时, 执行次序是首先将第 N 条次对角线 (即离主对角线最远) 上的元素化为 0, 然后再将 $N-1$ 条次对角线上的元素化为 0, 依此类推, 最后将第 2 条次对角线上的元素化为 0. 在消去第 k 条次对角线时, 是按行进行的, 即消去的元素的下标依次为

$$(k+1,1), \quad (k+2,2), \quad \ldots, \quad (n,n-k).$$

在消去元素 $(k+i,i)$ (左乘 Givens 变换) 的同时也消去元素 $(i,k+i)$ (右乘同一 Givens 变换的转置). 在消去元素 $(k+i,i)$ 时, 使用的 Givens 变换是 $G(k+i-1,k+i,\theta)$. 这样, 在消去元素 $(k+i,i)$ 和 $(i,k+i)$ 的同时, 会在 $(2k+i,k+i-1)$ 和 $(k+i-1,2k+i)$ 位置上出现非零元. 我们可以再通过 Givens 变换 $G(2k+i,2k+i-1,\theta)$ 将它们化为 0, 但此时会在

To be continued ...

参见蒋 < 矩阵计算 >

对称 QR 迭代算法的运算量

- 三对角化需要 $4n^3/3+O(n^2)$, 如果需要计算特征向量, 则运算量为 $8n^3/3+O(n^2)$;
- 对 T 做带位移的隐式 QR 迭代, 每次迭代的运算量为 $6n$;
- 计算 T 的特征值时, 假定每个平均迭代 2 步, 则计算所有特征值的运算量为 $12n^2$;
- 若要计算 T 的所有特征值和特征向量, 则运算量为 $6n^3+O(n^2)$;
- 若只要计算 A 的所有特征值, 运算量为 $4n^3/3+O(n^2)$;
- 若需要计算 A 的所有的特征值和特征向量, 则运算量为 $26n^3/3+O(n^2)$;

位移的选取 — Wilkinson 位移

位移的好坏直接影响到算法的收敛速度. 我们可以通过下面的方式来选取位移. 设

$$A^{(k)} = \begin{bmatrix} a_1^{(k)} & b_1^{(k)} & & \\ b_1^{(k)} & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1}^{(k)} \\ & & b_{n-1}^{(k)} & a_n^{(k)} \end{bmatrix},$$

一种简单的位移选取策略就是令 $\sigma_k = a_n^{(k)}$. 事实上, $a_n^{(k)}$ 就是收敛到特征向量的迭代向量的 Rayleigh 商. 这种位移选取方法几乎对所有的矩阵都有三次渐进收敛速度, 但也存在不收敛的例子, 故我们需要对其做一些改进.

☞ 如果在 QR 迭代算法中使用位移 $\sigma_k = a_n^{(k)}$, 而在 Rayleigh 商迭代算法 5.4 中取 $x_0 = e_n = [0, \ldots, 0, 1]^T$, 则利用 QR 迭代与反迭代之间的关系可以证明: QR 迭代算法中的 σ_k 与 Rayleigh 商迭代算法中的 ρ_k 相等.

一种有效的位移是 **Wilkinson 位移**, 即取 $\begin{bmatrix} a_{n-1}^{(k)} & b_{n-1}^{(k)} \\ b_{n-1}^{(k)} & a_n^{(k)} \end{bmatrix}$ 的最接近 $a_n^{(k)}$ 的特征值作为位移. 通过计算可得 Wilkinson 位移为

$$\sigma = a_n^{(k)} + \delta - \operatorname{sign}(\delta)\sqrt{\delta^2 + \left(b_{n-1}^{(k)}\right)^2}, \quad \text{其中} \quad \delta = \frac{1}{2}(a_{n-1}^{(k)} - a_n^{(k)}).$$

出于稳定性方面的考虑, 我们通常用下面的计算公式

$$\sigma = a_n^{(k)} - \frac{\left(b_{n-1}^{(k)}\right)^2}{\delta + \operatorname{sign}(\delta)\sqrt{\delta^2 + \left(b_{n-1}^{(k)}\right)^2}}.$$

定理 5.4 *采用 Wilkinson 位移的 QR 迭代是整体收敛的, 且至少是线性收敛. 事实上, 几乎对所有的矩阵都是渐进三次收敛的.*

证明. 见参考文献 [16, 29]. □

例 5.1 带 Wilkinson 位移的隐式 QR 迭代算法收敛性演示

(见 [9, page 214]), 相应的 Matlab 代码为 `TriQR.m`

5.5 分而治之 (Divide-and-Conquer)

分而治之算法在 1981 年由 Cuppen [8] 首次提出, 但直到 1995 年才出现稳定的实现方式 [19]. 该算法是目前计算维数大于 25 的矩阵的所有特征值和特征向量的最快算法. 下面我们介绍该算法.

考虑不可约对称三对角矩阵

$$
\begin{aligned}
T &= \left[\begin{array}{cccc|cccc}
a_1 & b_1 & & & & & & \\
b_1 & \ddots & \ddots & & & & & \\
& \ddots & a_{m-1} & b_{m-1} & & & & \\
& & b_{m-1} & a_m & b_m & & & \\ \hline
& & & b_m & a_{m+1} & b_{m+1} & & \\
& & & & b_{m+1} & \ddots & \ddots & \\
& & & & & \ddots & \ddots & b_{n-1} \\
& & & & & & b_{n-1} & a_n
\end{array}\right] \\
&= \left[\begin{array}{cccc|cccc}
a_1 & b_1 & & & & & & \\
b_1 & \ddots & \ddots & & & & & \\
& \ddots & a_{m-1} & b_{m-1} & & & & \\
& & b_{m-1} & a_m - b_m & & & & \\ \hline
& & & & a_{m+1} - b_m & b_{m+1} & & \\
& & & & b_{m+1} & \ddots & \ddots & \\
& & & & & \ddots & \ddots & b_{n-1} \\
& & & & & & b_{n-1} & a_n
\end{array}\right] + \left[\begin{array}{cc|cc}
& & & \\
& b_m & b_m & \\ \hline
& b_m & b_m & \\
& & &
\end{array}\right] \\
&= \left[\begin{array}{c|c} T_1 & 0 \\ \hline 0 & T_2 \end{array}\right] + b_m v v^T,
\end{aligned}
$$

其中 $v = [0, \dots, 0, 1, 1, 0, \dots, 0]^T$. 假定 T_1 和 T_2 的特征值分解已经计算出来了, 即 $T_1 = Q_1 \Lambda_1 Q_1^T$, $T_2 = Q_2 \Lambda_2 Q_2^T$, 下面考虑 T 的特征值分解.

首先介绍一个引理.

引理 5.3 设 $x, y \in \mathbb{R}^n$, 则 $\det(I + xy^T) = 1 + y^T x$.

证明. 见习题 5.3. □

我们首先考虑 T 的特征值与 T_1 和 T_2 的特征值之间的关系.

$$
\begin{aligned}
T &= \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} + b_m v v^T \\
&= \begin{bmatrix} Q_1 \Lambda_1 Q_1^T & 0 \\ 0 & Q_2 \Lambda_2 Q_2^T \end{bmatrix} + b_m v v^T \\
&= \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \left(\begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} + b_m u u^T \right) \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}^T,
\end{aligned}
$$

其中

$$u = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}^T v = \begin{bmatrix} Q_1^T \text{ 的最后一列} \\ Q_2^T \text{ 的第一列} \end{bmatrix}.$$

令 $\alpha = b_m, D = \mathrm{diag}(\Lambda_1, \Lambda_2) = \mathrm{diag}(d_1, d_2, \ldots, d_n)$, 并假设 $d_1 \geq d_2 \geq \cdots \geq d_n$. 则 T 的特征值与 $D + \alpha uu^T$ 的特征值相同.

下面计算 $D + \alpha uu^T$ 的特征值. 设 λ 是 $D + \alpha uu^T$ 的一个特征值, 若 $D - \lambda I$ 非奇异, 则

$$\det(D + \alpha uu^T - \lambda I) = \det(D - \lambda I) \cdot \det(I + \alpha(D - \lambda I)^{-1} uu^T).$$

故 $\det(I + \alpha(D - \lambda I)^{-1} uu^T) = 0$. 又由引理 5.3 可知

$$\det(I + \alpha(D - \lambda I)^{-1} uu^T) = 1 + \alpha u^T (D - \lambda I)^{-1} u = 1 + \alpha \sum_{i=1}^{n} \frac{u_i^2}{d_i - \lambda} \triangleq f(\lambda).$$

故求 A 的特征值等价于求**特征方程** (secular equation) $f(\lambda) = 0$ 的根. 由于

$$f'(\lambda) = \alpha \sum_{i=1}^{n} \frac{u_i^2}{(d_i - \lambda)^2},$$

当所有的 d_i 都互不相同, 且所有的 u_i 都不为零时, $f(\lambda)$ 在 $\lambda \neq d_i$ 处都是严格单调的 (见下图).

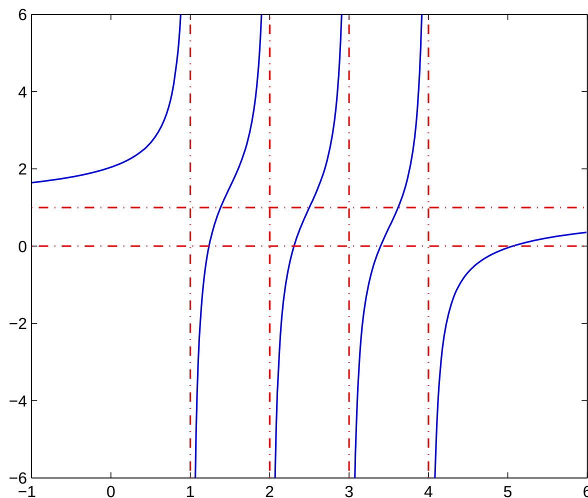


图 5.1 $f(\lambda) = 1 + 0.5 \left(\frac{1}{4-\lambda} + \frac{1}{3-\lambda} + \frac{1}{2-\lambda} + \frac{1}{1-\lambda} \right)$ 的图像

所以 $f(\lambda)$ 在每个区间 (d_i, d_{i+1}) 内都有一个根, 共 $n-1$ 个, 另一个根在 (d_1, ∞) (若 $\alpha > 0$) 或 $(-\infty, d_n)$ (若 $\alpha < 0$) 中. 由于 $f(\lambda)$ 在每个区间 (d_i, d_{i+1}) 内光滑且严格单调递增 ($\alpha > 0$) 或递减 ($\alpha < 0$), 所以在实际计算中, 可以使用对分法, 牛顿法及其变形, 或有理逼近等算法来求解. 通常都能很快收敛, 一般只需迭代几步即可. 因此, 计算一个特征值的运算量约为 $O(n)$, 计算 $D + \alpha uu^T$ 的所有特征值的运算量约为 $O(n^2)$.

当所有特征值计算出来后, 我们可以利用下面的引理来计算特征向量.

引理 5.4 设 $D \in \mathbb{R}^{n\times n}$ 为对角矩阵, $u \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$, 若 λ 是 $D+\alpha uu^T$ 的特征值, 则 $(D-\lambda I)^{-1}u$ 是其对应的特征向量, 且运算量为 $O(n)$.

证明. 由引理 5.3 可知

$$
\begin{aligned}
0 = \det(D+\alpha uu^T - \lambda I) &= \det(D-\lambda I)\cdot \det\left(I+\alpha(D-\lambda I)^{-1}uu^T\right) \\
&= \det(D-\lambda I)\cdot\left(1+\alpha u^T(D-\lambda I)^{-1}u\right),
\end{aligned}
$$

故 $1+\alpha u^T(D-\lambda I)^{-1}u=0$, 即 $\alpha u^T(D-\lambda I)^{-1}u=-1$. 直接计算可得

$$
\begin{aligned}
(D+\alpha uu^T)\left((D-\lambda I)^{-1}u\right) &= (D-\lambda I+\lambda I+\alpha uu^T)(D-\lambda I)^{-1}u \\
&= u+\lambda(D-\lambda I)^{-1}u+\left(\alpha u^T(D-\lambda I)^{-1}u\right)u \\
&= u+\lambda(D-\lambda I)^{-1}u-u \\
&= \lambda(D-\lambda I)^{-1}u,
\end{aligned}
$$

即引理结论成立. □

☞ 利用公式 $(D-\lambda I)^{-1}u$ 计算所有特征向量的运算量为 $O(n^2)$. 但这样计算是数值不稳定的.

算法 5.5 计算对称三对角矩阵的特征值和特征向量的分而治之法 (函数形式)

1: **function** $[Q,\Lambda]$ = dc_eig(T)   % $T=Q\Lambda Q^T$
2: **if** T is of 1×1 **then**
3:     $Q=1, \Lambda=T$
4:     return
5: **end if**
6: form $T=\begin{bmatrix} T_1 & 0 \\ 0 & T_2\end{bmatrix}+b_m vv^T$
7: $[Q_1,\Lambda_1]$ = dc_eig(T_1)
8: $[Q_2,\Lambda_2]$ = dc_eig(T_2)
9: form $D+\alpha uu^T$ from $\Lambda_1,\Lambda_2,Q_1,Q_2$
10: compute the eigenvalues Λ and eigenvectors $\hat{Q}$ of $D+\alpha uu^T$
11: compute the eigenvectors of T with $Q=\begin{bmatrix} Q_1 & 0 \\ 0 & Q_2\end{bmatrix}\cdot\hat{Q}$
12: end

下面我们详细讨论分而治之算法的几个细节问题:

(1) 如何减小运算量;
(2) 如何求解特征方程 $f(\lambda)=0$;
(3) 如何稳定地计算特征向量.

(1) 如何减小运算量 — 收缩技巧 (deflation)

分而治之算法的计算复杂性分析如下: 用 $t(n)$ 表示对 n 阶矩阵调用函数 `dc_eig` 的运算量, 则

$$\begin{aligned} t(n) = \ & 2\,t(n/2) && \text{递归调用 dc_eig 两次} \\ & + O(n^2) && \text{计算 } D+\alpha uu^T \text{ 的特征值和特征向量} \\ & + c\cdot n^3 && \text{计算 } Q. \end{aligned}$$

☞ 如果计算 Q 时使用的是稠密矩阵乘法, 则 $c = 2$;
若不计 $O(n^2)$ 项, 则由递归公式 $t(n) = 2\,t(n/2) + c\cdot n^3$ 可得 $t(n) \approx c\cdot 4n^3/3$.

但事实上, 由于**收缩** (deflation) 现象的存在, 常数 c 通常比 1 小得多.

在前面的算法描述过程中, 我们假定 d_i 互不相等且 u_i 不能为零. 事实上, 容易证明当 $d_i = d_{i+1}$ 或 $u_i = 0$ 时, d_i 即为 $D+\alpha uu^T$ 的特征值, 这种现象我们称为**收缩** (deflation) . 在实际计算时, 当 $d_i - d_{i+1}$ 或 $|u_i|$ 小于一个给定的阈值时, 我们就近似认为 d_i 为 $D+\alpha uu^T$ 的特征值, 即出现收缩现象.

在实际计算中, 收缩现象会经常发生, 而且非常频繁, 所以我们可以而且应该利用这种优点加快分而治之算法的速度 [8, 33].

由于主要的计算量集中在计算 Q, 即算法最后一步的矩阵乘积. 如果 $u_i = 0$, 则 d_i 为特征值, 其对应的特征向量为 e_i, 即 $\hat{Q}$ 的第 i 列为 e_i, 故计算 Q 的第 i 列时不需要做任何的计算. 当 $d_i = d_{i+1}$ 时, 也存在一个类似的简化.

(2) 特征方程求解

通常我们可以使用牛顿法来计算特征方程 $f(\lambda) = 0$ 的解.

当 $d_i \neq d_{i+1}$ 且 $u_i \neq 0$ 时, 我们用牛顿法计算 $f(\lambda)$ 在 (d_{i+1}, d_i) 中的零点 λ_i. 如果 $|u_i|$ 小于给定的阈值时, 我们可直接将 d_i 作为特征值 λ_i 的一个近似. 但当 u_i 很小 (却大于给定的阈值) 时, 此时 $f(\lambda)$ 在区间 $[d_{i+1}, d_i]$ 中的大部分处的斜率几乎为 0 (见图 5.2). 这时, 如果任取 $[d_{i+1}, d_i]$ 中的一个点作为迭代初始点, 经郭一次牛顿迭代后, 迭代解可能会跑到区间 $[d_{i+1}, d_i]$ 的外面, 造成不收敛.

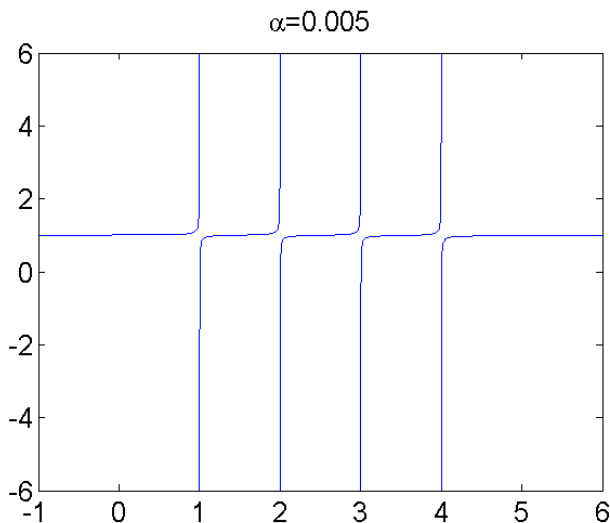


图 5.2 $f(\lambda)=1+0.005\left(\frac{1}{4-\lambda}+\frac{1}{3-\lambda}+\frac{1}{2-\lambda}+\frac{1}{1-\lambda}\right)$ 的图像

这时需要采用修正的牛顿法. 假设我们已经计算出 λ_i 的一个近似 $\tilde{\lambda}$, 下面我们需要从 $\tilde{\lambda}$ 出发, 利用牛顿迭代计算下一个近似, 直至收敛. 我们知道牛顿法的基本原理是使用 $f(\lambda)$ 在点 $\tilde{\lambda}$ 的切线来近似 $f(\lambda)$, 并将切线的零点作为下一个近似, 即用直线来近似曲线 $f(\lambda)$.

由于 u_i 很小时, 这种近似方法会出现问题, 所以我们不能使用直线来近似 $f(\lambda)$. 这时, 我们可以寻找其它简单函数 $h(\lambda)$ 来近似 $f(\lambda)$, 然后用 $h(\lambda)$ 的零点作为 $f(\lambda)$ 零点的近似, 并不断迭代下去, 直至收敛 (即更换迭代函数).

当然, $h(\lambda)$ 需要满足一定的要求: (1) 必须容易构造; (2) 其零点容易计算; (3) 尽可能与 $f(\lambda)$ 相近.

下面给出构造 $h(\lambda)$ 的一种方法. 因为 d_i 和 d_{i+1} 是 $f(\lambda)$ 的奇点, 所以我们令

$$h(\lambda)=\frac{c_1}{d_i-\lambda}+\frac{c_2}{d_{i+1}-\lambda}+c_3,$$

其中 c_1, c_2, c_3 为参数. 显然, $h(\lambda)$ 的零点很容易计算 (与 Newton 法相差无几). 在选取这些参数时, 要使得 $h(\lambda)$ 在 $\tilde{\lambda}$ 附近尽可能地接近 $f(\lambda)$. 记

$$f(\lambda)=1+\alpha\sum_{k=1}^{n}\frac{u_k^2}{d_k-\lambda}=1+\alpha\left(\sum_{k=1}^{i}\frac{u_k^2}{d_k-\lambda}+\sum_{k=i+1}^{n}\frac{u_k^2}{d_k-\lambda}\right)\triangleq 1+\alpha\big(\Psi_1(\lambda)+\Psi_2(\lambda)\big).$$

当 $\lambda\in(d_{i+1},d_i)$ 时, $\Psi_1(\lambda)$ 为正项的和, $\Psi_2(\lambda)$ 为负项的和, 因此它们都可以较精确地计算. 但如果把它们加在一起时可能会引起对消, 从而失去相对精度. 因此我们将 $h(\lambda)$ 写成

$$h(\lambda)=1+\alpha\big(h_1(\lambda)+h_2(\lambda)\big),$$

其中

$$h_1(\lambda)=\frac{c_1}{d_i-\lambda}+\hat{c}_1,\quad h_2(\lambda)=\frac{c_2}{d_{i+1}-\lambda}+\hat{c}_2$$

满足

$$h_1(\tilde{\lambda})=\Psi_1(\tilde{\lambda}),\quad h_1'(\tilde{\lambda})=\Psi_1'(\tilde{\lambda}),$$

$$h_2(\tilde{\lambda}) = \Psi_2(\tilde{\lambda}), \quad h_2'(\tilde{\lambda}) = \Psi_2'(\tilde{\lambda}).$$

即 $h_1(\lambda)$ 和 $h_2(\lambda)$ 分别在点 $\tilde{\lambda}$ 与 $\Psi_1(\lambda)$ 和 $\Psi_2(\lambda)$ 相切. 这在数值插值中是常见的条件. 容易计算可得

$$\begin{cases} c_1 = \Psi_1'(\tilde{\lambda})(d_i - \tilde{\lambda})^2, & \hat{c}_1 = \Psi_1(\tilde{\lambda}) - \Psi_1'(\tilde{\lambda})(d_i - \tilde{\lambda}), \\ c_2 = \Psi_2'(\tilde{\lambda})(d_{i+1} - \tilde{\lambda})^2, & \hat{c}_2 = \Psi_2(\tilde{\lambda}) - \Psi_2'(\tilde{\lambda})(d_{i+1} - \tilde{\lambda}). \end{cases} \tag{5.3}$$

所以, 最后取

$$h(\lambda) = 1 + \alpha(\hat{c}_1 + \hat{c}_2) + \alpha\left(\frac{c_1}{d_i - \lambda} + \frac{c_2}{d_{i+1} - \lambda}\right). \tag{5.4}$$

这就是迭代函数.

算法 5.6 修正的 Newton 算法

1: set $k = 0$
2: choose an initial guess $\lambda_0 \in [d_{i+1}, d_i]$
3: **while** not convergence **do**
4:     let $\tilde{\lambda} = \lambda_k$ and compute $c_1, c_2, \hat{c}_1, \hat{c}_2$ from (5.3)
5:     set $k = k + 1$
6:     compute the solution λ_k of $h(\lambda)$ defined by (5.4)
7: **end while**

---

(3) 计算特征向量的稳定算法

设 λ_i 是 $D + \alpha uu^T$ 的特征值, 则根据引理 5.4, 可利用公式 $(D - \lambda_i I)^{-1}u$ 来计算其对应的特征向量. 但遗憾的是, 当相邻的两个特征值非常接近时, 这个公式可能不稳定. 即当 λ_i 与 λ_{i+1} 非常接近时, 它们都靠近 d_{i+1} (这里假定 $\lambda_i \in (d_{i+1}, d_i)$), 在计算 $d_{i+1} - \lambda_i$ 和 $d_{i+1} - \lambda_{i+1}$ 时会存在对消, 这就可能损失有效数字, 产生较大的相对误差, 从而导致 $(D - \lambda_i I)^{-1}u$ 与 $(D - \lambda_{i+1} I)^{-1}u$ 的计算是不准确的, 正交性也会失去. 下面的定理可以解决这个问题. 详情参见 [17, 48].

定理 5.5 (Löwner) 设对角阵 $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$ 满足 $d_1 > d_2 > \cdots > d_n$. 若 $\lambda_1 > \lambda_2 > \cdots > \lambda_n$ 满足交错性质

$$\lambda_1 > d_1 > \lambda_2 > d_2 > \cdots > \lambda_n > d_n,$$

则存在一个向量 $\hat{u}$, 使得 $\lambda_1, \lambda_2, \ldots, \lambda_n$ 是 $\hat{D} = D + \hat{u}\hat{u}^T$ 的精确特征值, 其中 $\hat{u}$ 的分量由下面的公式给出

$$|\hat{u}_i| = \left(\frac{\prod_{k=1}^{n}(\lambda_k - d_i)}{\prod_{k=1,\, k\neq i}^{n}(d_k - d_i)}\right)^{1/2}.$$

证明. 由引理 5.3 可知 $\hat{D}$ 的特征多项式为

$$\begin{aligned} \det(\hat{D} - \lambda I) &= \det(D - \lambda I + \hat{u}\hat{u}^T) \\ &= \det(D - \lambda I) \cdot \det(I + (D - \lambda I)^{-1}\hat{u}\hat{u}^T) \\ &= \det(D - \lambda I) \cdot (1 + \hat{u}^T(D - \lambda I)^{-1}\hat{u}) \end{aligned}$$

$$
\begin{aligned}
&= \left(\prod_{k=1}^{n}(d_k-\lambda)\right)\cdot\left(1+\sum_{k=1}^{n}\frac{\hat{u}_k^2}{d_k-\lambda}\right) \\
&= \left(\prod_{k=1}^{n}(d_k-\lambda)\right)\cdot\left(1+\sum_{k=1,k\neq i}^{n}\frac{\hat{u}_k^2}{d_k-\lambda}\right)+\prod_{k=1}^{n}(d_k-\lambda)\hat{u}_i^2.
\end{aligned}
$$

又 $\lambda_1,\lambda_2,\ldots,\lambda_n$ 是 $\hat{D}$ 的特征值, 故

$$
\det(\hat{D}-\lambda I)=\prod_{k=1}^{n}(\lambda_k-\lambda).
$$

取 $\lambda=d_i$, 由 $\det(\hat{D}-\lambda I)$ 的两个表达式可得

$$
\prod_{k=1}^{n}(d_k-d_i)\hat{u}_i^2=\prod_{k=1}^{n}(\lambda_k-d_i),
$$

即

$$
\hat{u}_i^2=\frac{\prod_{k=1}^{n}(\lambda_k-d_i)}{\prod_{k=1,\,k\neq i}^{n}(d_k-d_i)}.
$$

由交错性质可知, 上式右边是正的, 故定理结论成立. □

下面给出计算矩阵 $D+uu^T$ 的特征值和特征向量的稳定算法.

算法 5.7 计算矩阵 $D+uu^T$ 的特征值和特征向量的稳定算法

1: Compute the eigenvalues $\lambda_1,\lambda_2,\ldots,\lambda_n$ by solving $f(\lambda)=0$
2: Compute $\hat{u}_i$ by Löwner Theorem so that $\lambda_1,\lambda_2,\ldots,\lambda_n$ are the exact eigenvalues of $D+\hat{u}\hat{u}^T$
3: Compute the eigenvectors of $D+\hat{u}\hat{u}^T$ by Lemma 5.4

通过分析可以说明上述算法计算出来的 $D+\hat{u}\hat{u}^T$ 的特征值和特征值向量是非常精确的, 它们与原矩阵 $D+uu^T$ 的特征值和特征向量非常相近, 这意味着该算法是数值稳定的.

箭型分而治之法

分而治之算法于 1981 年被首次提出, 但直到 1995 年才由 Gu 和 Eisenstat 给出了一种快速稳定的实现方式, 称为**箭型分而治之法** (Arrowhead Divide-and-Conquer , ADC). 他们做了大量的数值试验, 在试验中, 当矩阵规模不超过 6 时, 就采用对称 QR 迭代来计算特征值和特征向量. 在对特征方程求解时, 他们采用的是修正的有理逼近法. 数值结果表明, ADC 算法的计算精度可以与其他算法媲美, 而计算速度通常比对称 QR 迭代快 5 至 10 倍, 比 Cuppen 的分而治之法快 2 倍. 详细介绍见 [18, 19].

5.6 对分法和逆迭代 (Bisection and Inverse Iteration)

对分法的基本思想是利用惯性定理来计算所需的部分特征值.

定义 5.1 设 A 为对称矩阵, 则其**惯性**定义为

$$\text{Inertia}(A) = (\nu, \zeta, \pi)$$

其中 ν, ζ, π 分别表示 A 的负特征值, 零特征值和正特征值的个数.

定理 5.6 (Sylvester 惯性定理) 设 $A \in \mathbb{R}^{n\times n}$ 是对称矩阵, $X \in \mathbb{R}^{n\times n}$ 非奇异, 则 X^TAX 与 A 有相同的惯性.

利用 LU 分解可得 $A - zI = LDL^T$, 其中 L 为非奇异下三角矩阵, D 为对角阵, 则

$$\text{Inertia}(A - zI) = \text{Inertia}(D).$$

由于 D 是对角矩阵, 所以 $\text{Inertia}(D)$ 很容易计算.

设 $\alpha \in \mathbb{R}$, 记 $\text{Negcount}(A, \alpha)$ 为小于 α 的 A 的特征值的个数, 即

$$\text{Negcount}(A, \alpha) = \#(\lambda(A) < \alpha).$$

设 $\alpha_1 < \alpha_2$, 则 A 在区间 $[\alpha_1, \alpha_2)$ 中的特征值个数为

$$\text{Negcount}(A, \alpha_2) - \text{Negcount}(A, \alpha_1).$$

如果 $\alpha_2 - \alpha_1 < tol$ (其中 $tol \ll 1$ 为事先给定的阈值), 且 A 在 $[\alpha_1, \alpha_2)$ 中有特征值, 则我们可将 $[\alpha_1, \alpha_2)$ 中的任意一个值作为 A 在该区间中的特征值的近似.

由此我们可以给出下面的对分法.

算法 5.8 计算 A 在 $[a, b)$ 中的所有特征值

1: Let tol be a given threshold
2: compute $n_a = \text{Negcount}(A, a)$
3: compute $n_b = \text{Negcount}(A, b)$
4: **if** $n_a = n_b$ **then**
5:     return   % 此时 $[a, b)$ 中没有 A 的特征值
6: **end if**
7: put (a, n_a, b, n_b) onto $worklist$
8:   % $worklist$ 中的元素是“四元素对”, 即由四个数组成的数对
9: **while** $worklist$ not empty **do**
10:     remove $(low, n_{low}, up, n_{up})$ from the $worklist$
11:   % $(low, n_{low}, up, n_{up})$ 是 $worklist$ 中的任意一个元素
12:     **if** $(up - low) < tol$ **then**
13:         print "There are $n_{up} - n_{low}$ eigenvalues in [low, up)"
14:     **else**

15: compute $mid = (low + up)/2$
16: compute $n_{mid} = \text{Negcount}(A, mid)$
17: **if** $(n_{mid} > n_{low})$ **then**
18: put $(low, n_{low}, mid, n_{mid})$ onto $worklist$
19: **end if**
20: **if** $(n_{up} > n_{mid})$ **then**
21: put $(mid, n_{mid}, up, n_{up})$ onto $worklist$
22: **end if**
23: **end if**
24: **end while**

显然, 对分法的主要运算量集中在计算 Negcount(A, z). 如果 A 是稠密的, 则可用选主元的对称 Gauss 消元法来计算, 但此时的运算量为 $O(n^3)$, 显然不实用. 但如果事先将 A 转化成对称三对角矩阵, 计算 $A - zI$ 的 LDL^T 分解非常简单:

$$A - zI = \begin{bmatrix} a_1 - z & b_1 & & \\ b_1 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n - z \end{bmatrix}$$

$$= \begin{bmatrix} 1 & & & \\ l_1 & \ddots & & \\ & \ddots & \ddots & \\ & & l_{n-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \begin{bmatrix} 1 & l_1 & & \\ & \ddots & \ddots & \\ & & \ddots & l_{n-1} \\ & & & 1 \end{bmatrix} \triangleq LDL^T.$$

利用待定系数法, 可以得到下面的递推公式

$$d_1 = a_1 - z, \quad d_i = (a_i - z) - \frac{b_{i-1}^2}{d_{i-1}}, \qquad i = 2, 3, \ldots, n. \tag{5.5}$$

用上面的公式计算 d_i 的运算量约为 $4n$.

注意这里没有选主元, 但针对对称三对角矩阵, 该算法是非常稳定的, 即使当 d_i 有可能很小时, 算法依然很稳定.

定理 5.7 利用公式 (5.5) 计算所得的 d_i 与精确计算 $\hat{A}$ 的 $\hat{d}_i$ 有相同的符号, 故有相同的惯性. 这里 $\hat{A}$ 与 A 非常接近, 即

$$\hat{A}(i, i) = a_i, \quad \hat{A}(i, i+1) = b_i(1 + \varepsilon_i),$$

其中 $|\varepsilon_i| \leq 2.5\varepsilon + O(\varepsilon^2)$, 这里的 ε 为机器精度.

证明. 略 (见教材第 194 页). □

- 由于单独调用一次 Negcount 的运算量为 $4n$, 故计算 k 个特征值的总运算量约为 $O(kn)$;
- 当特征值计算出来后, 我们可以使用带位移的逆迭代来计算对应的特征向量. 通常只需迭代 1 至 2 次即可, 由于 A 是三对角矩阵, 故计算每个特征向量的运算量为 $O(n)$;

- 当特征值紧靠在一起时, 计算出来的特征向量可能会失去正交性, 此时需要进行再正交化, 可通过 MGS 的 QR 分解来实现.

5.7 奇异值分解

奇异值分解 (SVD) 具有十分广泛的应用背景, 因此, 如何更好更快地计算一个给定矩阵的 SVD 是科学与工程计算领域中的一个热门研究课题, 吸引了众多专家进行这方面的研究, 也涌现出了许多奇妙的方法. 本章主要介绍计算 SVD 的常用算法.

对任意矩阵 $A \in \mathbb{R}^{m\times n}$, 其奇异值与对称矩阵 A^TA, AA^T 和 $\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$ 的特征值是是密切相关的, 故理论上计算对称特征值的算法都可以用于计算奇异值. 但在实际计算中, 我们通过可以利用 SVD 的特殊结构使得算法更加有效和准确.

与计算对称矩阵的特征值类似, 计算一个矩阵 A 的奇异值分解的算法通常分为以下几个步骤 (Jacobi 算法除外):

1. 将 A 二对角化: $B = U_1^TAV_1$, 其中 B 为上二对角矩阵, U_1, V_1 为正交阵;
2. 计算 B 的 SVD: $B = U_2\Sigma V_2^T$, 其中 Σ 为对角阵, U_2, V_2 为正交阵;
3. 合并得到 A 的 SVD: $A = U_1BV_1^T = (U_1U_2)B(V_1V_2)^T$.

5.7.1 二对角化

我们知道, 对称矩阵可以通过一系列 Householder 变换转化为对称三对角矩阵. 对于一般矩阵 $A \in \mathbb{R}^{m\times n}$, 我们也可以通过 Householder 变换, 将其转化为二对角矩阵, 即计算正交矩阵 U_1 和 V_1 使得

$$U_1^TAV_1 = B, \tag{5.6}$$

其中 B 是一个实 (上) 二对角矩阵. 这个过程就称为**二对角化**.

☞ 需要注意的是, 与对称矩阵的对称三对角化不同, A 与 B 是不相似的.

设 $A \in \mathbb{R}^{m\times n}$, 二对角化过程大致如下:

(1) 首先确定一个 Household 矩阵 $H_1 \in \mathbb{R}^{m\times m}$, 使得 H_1A 的第一列除第一个元素外, 其它分量都为零, 即

$$H_1A = \begin{bmatrix} * & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & * & * & \cdots & * \end{bmatrix}.$$

(2) 再确定一个 Household 矩阵 $\tilde{H}_1 \in \mathbb{R}^{(n-1)\times(n-1)}$, 把 H_1A 的第一行的第 3 至第 n 个元素化为零, 即

$$H_1A\begin{bmatrix} 1 & 0 \\ 0 & \tilde{H}_1 \end{bmatrix} = \begin{bmatrix} * & * & 0 & \cdots & 0 \\ 0 & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & * & * & \cdots & * \end{bmatrix}.$$

(3) 重复上面的过程, 直到把 A 最终化为二对角矩阵.

有了分解 (5.6) 以后, 我们可得

$$A^TA = (U_1BV_1^T)^TU_1BV_1^T = V_1B^TBV_1^T,$$

即 $V_1^TA^TAV_1 = B^TB$. 由于 B^TB 是对称三对角的, 所以这就相当于将 A^TA 三对角化.

整个二对角化过程的运算量约为 $4mn^2 + 4m^2n - 4n^3/3$, 若不需要计算 U_1 和 V_1, 则运算量约为 $4mn^2 - 4n^3/3$.

二对角矩阵的奇异值分解

设 $B \in \mathbb{R}^{n\times n}$ 是一个二对角矩阵

$$B = \begin{bmatrix} a_1 & b_1 & & \\ & \ddots & \ddots & \\ & & \ddots & b_{n-1} \\ & & & a_n \end{bmatrix},$$

则下面三种方法均可将计算 B 的 SVD 转化成计算对称三对角矩阵的特征分解:

(1) 令 $A = \begin{bmatrix} 0 & B^T \\ B & 0 \end{bmatrix}$, 置换阵 $P = [e_1, e_{n+1}, e_2, e_{n+2}, \ldots, e_n, e_{2n}]$, 则 $T_{ps} = P^TAP$ 是对称三对角矩阵, 且 T_{ps} 的主对角线元素全为 0, 次对角线元素为 $a_1, b_1, a_2, b_2, \ldots, a_{n-1}, b_{n-1}, a_n$. 若 (λ_i, x_i) 是 T_{ps} 的一个特征对, 则

$$\lambda_i = \pm\sigma_i, \quad Px_i = \frac{1}{\sqrt{2}}\begin{bmatrix} v_i \\ \pm u_i \end{bmatrix},$$

其中 σ_i 为 B 一个奇异值, u_i 和 v_i 分别为对应的左和右奇异向量.

(2) 令 $T_{BB^T} = BB^T$, 则

$$T_{BB^T} = \begin{bmatrix} a_1^2+b_1^2 & a_2b_1 & & \\ a_2b_1 & \ddots & \ddots & \\ & \ddots & a_{n-1}^2+b_{n-1}^2 & a_nb_{n-1} \\ & & a_nb_{n-1} & a_n^2 \end{bmatrix}.$$

T_{BB^T} 的特征值为 B 的奇异值的平方, 且 T_{BB^T} 的特征向量为 B 的左奇异向量.

(3) 令 $T_{B^TB} = B^TB$, 则

$$T_{B^TB} = \begin{bmatrix} a_1^2 & a_1b_1 & & \\ a_1b_1 & a_2^2+b_1^2 & \ddots & \\ & \ddots & \ddots & a_{n-1}b_{n-1} \\ & & a_{n-1}b_{n-1} & a_n^2+b_{n-1}^2 \end{bmatrix}.$$

T_{B^TB} 的特征值为 B 的奇异值的平方, 且 T_{B^TB} 的特征向量为 B 的右奇异向量.

理论上, 我们可以直接使用 QR 迭代、分而治之法或带反迭代的对分法, 计算三对角矩阵 T_{ps}, T_{BB^T} 和 T_{B^TB} 的特征值和特征向量. 但一般来说, 这种做法并不是最佳的, 原因如下:

(1) 对 T_{ps} 做 QR 迭代并不划算, 因为 QR 迭代计算所有的特征值和特征向量, 而事实上只要计算正的特征值即可;

(2) 直接构成 T_{BB^T} 或 T_{B^TB} 是数值不稳定的. 事实上, 这样做可能会使得 B 的小奇异值的精度丢失一半.

下面是一些计算奇异值分解的比较实用的算法.

1. **Golub-Kahan SVD 算法**: 由 Golub 和 Kahan [15] 于 1965 年提出, 是一种十分稳定且高效的计算 SVD 的算法. 主要思想是将带位移的对称 QR 迭代算法隐式地用到 B^TB 上, 在该算法中, 并不需要显示地把 B^TB 计算出来. 该算法也通常就称为 SVD 算法, 是一个基本且实用的算法, 目前仍然是计算小规模矩阵奇异值分解的常用算法. 关于这个算法的详细描述, 可参见 [16].
2. **dqds 算法**: 由 Fernando 和 Parlett [12] 于 1994 年提出, 是计算二对角矩阵所有奇异值的最快算法, 而且能达到很高的相对精度, 包括奇异值很小的情形. 该算法主要基于对 B^TB 的 Cholesky 迭代, 可以看作是 LR 迭代算法的改进. 由于 LR 迭代算法在一定条件下与对称 QR 算法是等价的, 因此该算法也可以看作是 QR 迭代的变形.
3. **分而治之法**: 该算法是计算维数 $n \geq 25$ 的矩阵的所有奇异值和奇异向量的最快算法, 但不能保证小奇异值的相对精度, 即 σ_i 的相对精度为 $O(\varepsilon)\sigma_1$, 而不是 $O(\varepsilon)\sigma_i$.
4. **对分法和反迭代**: 主要用于计算某个区间内的奇异值及对应的奇异向量, 能保证较高的相对精度.
5. **Jacobi 迭代**: 可隐式地对 AA^T 或 A^TA 实施对称 Jacobi 迭代, 能保证较高的相对精度. 最近, Z. Drmač 和 K. Veselić [10, 11] 改进了最初的 Jacobi 算法, 使其变成一个速度快、精度高的实用算法.

在这里, 我们介绍 Golub-Kahan SVD 算法, dqds 算法和 Jacobi 迭代.

5.7.2 Golub-Kahan SVD 算法

参见 [53] *To be continued ...*

该算法主要思想是将带位移的对称 QR 迭代算法隐式地用到 B^TB 上, 而无需将 B^TB 显式地计算出来.

算法基本框架

Golub-Kahan SVD 算法有时也简称 SVD 算法, 其基本框架是:

- 将矩阵 A 二对角化, 得到上二对角矩阵 B;
- 用隐式 QR 迭代计算 B^TB 的特征值分解, 即

$$B^TB = Q\Lambda Q^T, \quad \Lambda = \operatorname{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2). \tag{5.7}$$

- 计算 BQ 的列主元 QR 分解, 即

$$(BQ)P = UR, \tag{5.8}$$

其中 P 是置换矩阵, U 是正交矩阵, R 是上三角矩阵.

由 (5.7) 可知

$$(BQ)^TBQ = \Lambda,$$

因此 BQ 是列正交矩阵 (但不是单位列正交). 再由 (5.8) 可知 $R = U^T(BQ)P$ 也是列正交矩阵. 又 R 是上三角矩阵, 所以 R 必定是对角矩阵. 令 $V = QP$, 则由 (5.8) 可知

$$U^T BV = R.$$

这就是二对角矩阵 B 的奇异值分解.

算法的具体实现

首先是选取位移, 即 $B^T B$ 右下角 2×2 矩阵的特征值. 易知该 2×2 矩阵为

$$\begin{bmatrix} a_{n-1}^2 + b_{n-2}^2 & a_{n-1}b_{n-1} \\ a_{n-1}b_{n-1} & a_n^2 + b_{n-1}^2 \end{bmatrix}.$$

将其靠近 $a_n^2 + b_{n-1}^2$ 的特征值作为位移 σ.

5.7.3 dqds 算法

我们首先介绍针对实对称正定矩阵的 LR 算法, 该算法思想与 QR 迭代算法类似, 但提出时间更早.

算法 5.9 带位移的 LR 算法

1: Let T_0 be a given real symmetric positive definite matrix
2: set $i = 0$
3: **while** not converge **do**
4: choose a shift τ_i^2 satisfying $\tau_i^2 < \min\{\lambda(T_i)\}$
5: compute B_i such that $T_i - \tau_i^2 I = B_i^T B_i$ % Cholesky factorization
6: $T_{i+1} = B_i B_i^T + \tau_i^2 I$
7: $i = i + 1$
8: **end while**

LR 迭代算法在形式上与 QR 迭代算法非常类似. 事实上, 对于不带位移的 LR 迭代算法, 我们可以证明, 两步 LR 迭代等价于一步 QR 迭代.

引理 5.5 *设 $\tilde{T}$ 是不带位移的 LR 算法迭代两步后生成的矩阵, $\hat{T}$ 是不带位移的 QR 算法迭代一步后生成的矩阵, 则 $\tilde{T} = \hat{T}$.*

证明. *To be continued ...* □

☞ (1) LR 算法中要求 T_0 对称正定, 但并不一定是三对角矩阵;
(2) 由该引理可知, QR 算法与 LR 算法有相同的收敛性.

下面我们介绍 dqds 算法. 该算法是针对三对角的对称正定矩阵 $B^T B$, 其中 B 是二对角矩阵. 在数学上, dqds 算法与 LR 算法是等价的, 但在该算法中, 我们是直接通过 B_i 来计算 B_{i+1}, 从而避免计算中间矩阵 T_{i+1}, 这样也就尽可能地避免了由于计算 $B_i B_i^T$ 而可能带来的数值不稳定性.

下面推导如何从 B_i 直接计算 B_{i+1}. 设

$$
B_i = \begin{bmatrix} a_1 & b_1 & & \\ & a_2 & \ddots & \\ & & \ddots & b_{n-1} \\ & & & a_n \end{bmatrix}, \quad B_{i+1} = \begin{bmatrix} \tilde{a}_1 & \tilde{b}_1 & & \\ & \tilde{a}_2 & \ddots & \\ & & \ddots & \tilde{b}_{n-1} \\ & & & \tilde{a}_n \end{bmatrix}.
$$

为了书写方便, 我们记 $b_0 = b_n = \tilde{b}_0 = \tilde{b}_n = 0$. 由 LR 算法 5.9 可知

$$
B_{i+1}^T B_{i+1} + \tau_{i+1}^2 I = B_i B_i^T + \tau_i^2 I.
$$

比较等式两边矩阵的对角线和上对角线元素, 可得

$$
\begin{aligned}
&\tilde{a}_k^2 + \tilde{b}_{k-1}^2 + \tau_{i+1}^2 = a_k^2 + b_k^2 + \tau_i^2, \quad k = 1, 2, \ldots, n \\
&\tilde{a}_k \tilde{b}_k = a_{k+1} b_k \quad \text{或} \quad \tilde{a}_k^2 \tilde{b}_k^2 = a_{k+1}^2 b_k^2, \quad k = 1, 2, \ldots, n-1.
\end{aligned}
$$

记 $\delta = \tau_{i+1}^2 - \tau_i^2$, $p_k = a_k^2$, $q_k = b_k^2$, $\tilde{p}_k = \tilde{a}_k^2$, $\tilde{q}_k = \tilde{b}_k^2$, 则可得下面的 qds 算法.

算法 5.10 qds 算法的单步 ($B_i \to B_{i+1}$)

1: $\delta = \tau_{i+1}^2 - \tau_i^2$
2: **for** $k = 1$ to $n-1$ **do**
3: $\quad \tilde{p}_k = p_k + q_k - \tilde{q}_{k-1} - \delta$
4: $\quad \tilde{q}_k = q_k \cdot (p_{k+1}/\tilde{p}_k)$
5: **end for**
6: $\tilde{p}_n = p_n - \tilde{q}_{n-1} - \delta$

qds 算法中的每个循环仅需 5 个浮点运算, 所以运算量较少. 为了提高算法的精确性, 我们引入一个辅助变量 $d_k \triangleq p_k - \tilde{q}_{k-1} - \delta$, 则

$$
\begin{aligned}
d_k &= p_k - \tilde{q}_{k-1} - \delta \\
&= p_k - \frac{q_{k-1} p_k}{\tilde{p}_{k-1}} - \delta \\
&= p_k \cdot \frac{\tilde{p}_{k-1} - q_{k-1}}{\tilde{p}_{k-1}} - \delta \\
&= p_k \cdot \frac{p_{k-1} - \tilde{q}_{k-2} - \delta}{\tilde{p}_{k-1}} - \delta \\
&= \frac{p_k}{\tilde{p}_{k-1}} \cdot d_{k-1} - \delta.
\end{aligned}
$$

于是就可得到 dqds 算法 5.11.

算法 5.11 dqds 算法的单步 ($B_i \to B_{i+1}$)

1: $\delta = \tau_{i+1}^2 - \tau_i^2$
2: $d_1 = p_1 - \delta$
3: **for** $k = 1$ to $n-1$ **do**

4: $\tilde{p}_k = d_k + q_k$
5: $t = p_{k+1}/\tilde{p}_k$
6: $\tilde{q}_j = q_k \cdot t$
7: $d_{k+1} = d_k \cdot t - \delta$
8: **end for**
9: $\tilde{p}_n = d_n$

dqds 算法的运算量与 dqs 差不多, 但更精确. 这里我们只列出相应的结果.

引理 5.6 设二对角矩阵 B 的对角元和上对角元分别为 $a_1, a_2, \ldots, a_n$ 和 $b_1, b_2, \ldots, b_{n-1}$, $\tilde{B}$ 的对角元和上对角元分别为 $\tilde{a}_i = a_i\eta_i$, $\tilde{b}_i = b_i\xi_i$. 则 $\tilde{B} = D_1 B D_2$, 其中

$$D_1 = \mathrm{diag}\left(\eta_1, \frac{\eta_2\eta_1}{\xi_1}, \frac{\eta_3\eta_2\eta_1}{\xi_2\xi_1}, \ldots, \frac{\eta_n\eta_{n-1}\cdots\eta_1}{\xi_{n-1}\xi_{n-2}\cdots\xi_1}\right),$$
$$D_2 = \mathrm{diag}\left(1, \frac{\xi_1}{\eta_1}, \frac{\xi_2\xi_1}{\eta_2\eta_1}, \ldots, \frac{\xi_{n-1}\xi_{n-2}\cdots\xi_1}{\eta_{n-1}\eta_{n-2}\cdots\eta_1}\right).$$

定理 5.8 设 B 和 $\tilde{B}$ 的定义如引理 5.6. 若存在 $\tau > 1$ 使得 $\tau^{-1} \leq \eta_i \leq \tau$, $\tau^{-1} \leq \xi_i \leq \tau$, 即 $\varepsilon = \tau - 1$ 是 B 与 $\tilde{B}$ 的相应元素之间的相对误差的一个上界. 设 B 与 $\tilde{B}$ 的奇异值分别为 $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ 和 $\tilde{\sigma}_1 \geq \tilde{\sigma}_2 \geq \ldots \geq \tilde{\sigma}_n$, 则

$$\left|\tilde{\sigma}_i - \sigma_i\right| \leq (\tau^{4n-2} - 1)\sigma_i.$$

若 $\sigma_i \neq 0$, $\varepsilon = \tau - 1 \ll 1$, 则上式可记为

$$\frac{\left|\tilde{\sigma}_i - \sigma_i\right|}{\sigma_i} \leq \tau^{4n-2} - 1 = (4n-2)\varepsilon + O(\varepsilon^2).$$

下面给出 dqds 算法的高精度性.

定理 5.9 以浮点运算对 B 应用 *dqds* 算法的单步, 得到矩阵 $\tilde{B}$, 该过程等价于

1. 对 B 的每个元素作一个小的相对扰动 (不超过 1.5ε), 得到 $\tilde{B}$;
2. 对 $\tilde{B}$ 应用精确的 *dqds* 算法的单步, 得到 $\bar{B}$;
3. 对 $\bar{B}$ 的每个元素作一个小的相对扰动 (不超过 ε), 得到 $\tilde{B}$.

由定理 5.8 可知, B 和 $\tilde{B}$ 的奇异值满足高的相对精度.

关于 dqds 算法中位移的选取, 以及如何判断收敛性, 可以参见 [12].

5.7.4 Jacobi 算法

本节讨论对矩阵 $M = A^TA$ 实施隐式的 Jacobi 算法来计算 A 的奇异值.

我们知道, Jaboci 算法的每一步就是对矩阵作 Jacobi 旋转, 即 $A^TA \to J^TA^TAJ$, 其中 J 的选取将某两个非对角元化为 0. 在实际计算中, 我们只需计算 AJ, 故该算法称为**单边 Jacobi 旋转**.

算法 5.12 单边 Jacobi 旋转的单步

% 对 $M = A^TA$ 作 Jacobi 旋转, 将 $M(i,j), M(j,i)$ 化为 0

1: Compute $m_{ii} = (A^TA)_{ii}, m_{ij} = (A^TA)_{ij}, m_{jj} = (A^TA)_{jj}$
2: **if** m_{ij} is not small enough **then**
3: $\quad \tau = (m_{ii} - m_{jj})/(2 \cdot m_{ij})$
4: $\quad t = \text{sign}(\tau)/(|\tau| + \sqrt{1+\tau^2})$
5: $\quad c = 1/\sqrt{1+t^2}$
6: $\quad s = c \cdot t$
7: $\quad A = AG(i,j,\theta)$ % $G(i,j,\theta)$ 为 Givens 变换
8: $\quad$ **if** eigenvectors are desired **then**
9: $\quad\quad J = J \cdot G(i,j,\theta)$
10: $\quad$ **end if**
11: **end if**

在上面算法的基础上, 我们可以给出完整的单边 Jacobi 算法.

算法 5.13 单边 Jacobi 算法: 计算 $A = U\Sigma V^T$

1: **while** A^TA is not diagonal enough **do**
2: $\quad$ **for** $i = 1$ to $n-1$ **do**
3: $\quad\quad$ **for** $j = i+1$ to n **do**
4: $\quad\quad\quad$ 调用单边 Jacobi 旋转
5: $\quad\quad$ **end for**
6: $\quad$ **end for**
7: **end while**
8: compute $\sigma_i = \|A(:,i)\|_2, i = 1, 2, \ldots n$
9: $U = [u_1, \ldots, u_n]$ with $u_i = A(:,i)/\sigma_i$
10: $V = J$

Jacobi 算法的特点:

- 不需要双对角化, 这样可以避免双对角化引入的误差;
- 可达到相对较高的计算精度;
- 速度较慢. (目前已有快速的改进算法, 参见 [10, 11])

定理 5.10 设 $A = DX \in \mathbb{R}^{n\times n}$, 其中 D 为非奇异对角阵, X 非奇异. 设 $\hat{A}$ 是按浮点运算单边 Jacobi 旋转 m 次后所得到的矩阵. 若 A 和 $\hat{A}$ 的奇异值分别为 $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ 和 $\hat{\sigma}_1 \geq \hat{\sigma}_2 \geq \ldots \geq \hat{\sigma}_n$, 则

$$\frac{|\hat{\sigma}_i - \sigma_i|}{\sigma_i} \leq O(m\varepsilon)\kappa(X).$$

故 X 的条件数越小, 计算矩阵 A 的奇异值时相对误差越小.

5.8 扰动分析

设 $A \in \mathbb{R}^{n\times n}$ 是对称矩阵, 则有下面的谱分解.

定理 5.11 设 $A \in \mathbb{R}^{n\times n}$ 是对称矩阵. 则存在一个正交矩阵 Q 使得

$$A = Q\Lambda Q^T$$

其中 $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ 是一个实对角矩阵.

这里的 λ_i 就是 A 的特征值, 我们假设 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. 令 $Q = [q_1, q_2, \dots, q_n]$, 则 q_i 就是 λ_i 对应的单位正交特征向量.

关于对称矩阵特征值问题的扰动理论, 这里只做一些简单介绍, 若要深入了解这方面的信息, 可以参考 [22, 23, 38, 50].

5.8.1 特征值与 Rayleigh 商

定义 5.2 设 $A \in \mathbb{R}^{n\times n}$ 是对称矩阵, 向量 $x \in \mathbb{R}^n$ 非零, 则 x 关于 A 的 Rayleigh 商定义为:

$$\rho(x, A) = \frac{x^T Ax}{x^T x}.$$

有时简记为 $\rho(x)$.

下面是关于 Rayleigh 商的一些基本性质:

(1) $\rho(\alpha x) = \rho(x), \forall \alpha \in \mathbb{R}, \alpha \neq 0$;

(2) $\rho(q_i) = \lambda_i, i = 1, 2, \dots, n$;

(3) 设 $x = \alpha_1 q_1 + \alpha_2 q_2 + \cdots + \alpha_n q_n$, 则

$$\rho(x) = \frac{\alpha_1^2\lambda_1 + \alpha_2^2\lambda_2 + \cdots + \alpha_n^2\lambda_n}{\alpha_1^2 + \alpha_2^2 + \cdots + \alpha_n^2};$$

(4) $\lambda_n \leq \rho(x) \leq \lambda_1, |\rho(x)| \leq \|A\|_2$.

实对称矩阵的特征值与 Rayleigh 商之间的一个基本性质是 Courant-Fischer 极小极大定理.

定理 5.12 (Courant-Fischer) 设 $A \in \mathbb{R}^{n\times n}$ 是对称矩阵, 其特征值为 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, 则有

$$\lambda_k = \max_{\mathbb{U}\in\mathbb{S}_k^n} \min_{x\in\mathbb{U},\, x\neq 0} \frac{x^T Ax}{x^T x} = \min_{\mathbb{V}\in\mathbb{S}_{n-k+1}^n} \max_{x\in\mathbb{V},\, x\neq 0} \frac{x^T Ax}{x^T x},$$

其中 $\mathbb{S}_i^n$ 表示 $\mathbb{R}^n$ 中所有 i 维子空间构成的集合. 当

$$\mathbb{U} = \mathrm{span}\{q_1, \dots, q_k\}, \quad \mathbb{V} = \mathrm{span}\{q_k, \dots, q_n\}, \quad x = q_k$$

时, 上式中的等号成立.

证明. 设 $\mathbb{U} \in \mathbb{S}_k^n$ 和 $\mathbb{V} \in \mathbb{S}_{n-k+1}^n$ 分别为 $\mathbb{R}^n$ 中任意的 k 和 $n-k+1$ 维子空间. 由于

$$\dim \mathbb{U} + \dim \mathbb{V} = n + 1 > n,$$

可得

$$\mathbb{U} \cap \mathbb{V} \neq \{0\}.$$

故存在非零向量 $\tilde{x} \in \mathbb{U} \cap \mathbb{V}$. 所以有

$$\min_{x \in \mathbb{U},\, x \neq 0} \rho(x) \leq \rho(\tilde{x}) \leq \max_{x \in \mathbb{V},\, x \neq 0} \rho(x)\ .$$

由 $\mathbb{U}$ 和 $\mathbb{V}$ 的任意性可知,

$$\max_{\mathbb{U} \in \mathbb{S}_k^n} \min_{x \in \mathbb{U},\, x \neq 0} \rho(x) \leq \min_{\mathbb{V} \in \mathbb{S}_{n-k+1}^n} \max_{x \in \mathbb{V},\, x \neq 0} \rho(x)\ . \tag{5.9}$$

取 $\mathbb{U} = \mathrm{span}\{q_1, \ldots, q_k\}$, 则 $\mathbb{U}$ 中的任意向量都可写成 $x = \alpha_1 q_1 + \cdots + \alpha_k q_k$, 此时

$$\rho(x) = \frac{x^T A x}{x^T x} = \frac{\alpha_1^2 \lambda_1 + \cdots + \alpha_k^2 \lambda_k}{\alpha_1^2 + \cdots + \alpha_k^2} \geq \frac{\sum_{i=1}^k \alpha_i^2 \lambda_k}{\sum_{i=1}^k \alpha_i^2} = \lambda_k\ ,$$

即

$$\max_{\mathbb{U} \in \mathbb{S}_k^n} \min_{x \in \mathbb{U},\, x \neq 0} \rho(x) \geq \lambda_k. \tag{5.10}$$

同理, 取 $\mathbb{V} = \mathrm{span}\{q_k, \ldots, q_n\}$, 则 $\mathbb{V}$ 中的任意向量都可写成 $x = \alpha_k q_k + \cdots + \alpha_n q_n$, 此时

$$\rho(x) = \frac{x^T A x}{x^T x} = \frac{\alpha_k^2 \lambda_k + \cdots + \alpha_n^2 \lambda_n}{\alpha_k^2 + \cdots + \alpha_n^2} \leq \frac{\sum_{i=k}^n \alpha_i^2 \lambda_k}{\sum_{i=k}^n \alpha_i^2} = \lambda_k,$$

即

$$\min_{\mathbb{V} \in \mathbb{S}_{n-k+1}^n} \max_{x \in \mathbb{V},\, x \neq 0} \rho(x) \leq \lambda_k. \tag{5.11}$$

由 (5.9), (5.10), (5.11) 可知, 定理结论成立. □

当 $k = 1$ 和 $k = n$ 时, 就可以得到下面的定理.

定理 5.13 (Rayleigh-Ritz) 设 $A \in \mathbb{R}^{n \times n}$ 是对称矩阵, 其特征值为 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, 则有

$$\lambda_1 = \max_{x \in \mathbb{R}^n,\, x \neq 0} \frac{x^T A x}{x^T x}, \quad \lambda_n = \min_{x \in \mathbb{R}^n,\, x \neq 0} \frac{x^T A x}{x^T x}.$$

由极小极大定理, 我们可以得到下面的特征值分隔定理.

定理 5.14 (分隔定理) 设 $A \in \mathbb{R}^{n \times n}$ 是对称矩阵, $B = Q^T A Q$, 其中 $Q \in \mathbb{R}^{n \times (n-1)}$ 满足 $Q^T Q = I_{n-1}$. 再设 A 和 B 的特征值分别为

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \quad 和 \quad \tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_{n-1},$$

则有

$$\lambda_1 \geq \tilde{\lambda}_1 \geq \lambda_2 \geq \tilde{\lambda}_2 \cdots \geq \tilde{\lambda}_{n-1} \geq \lambda_n.$$

特别地, 在定理 5.14 中, 取 $Q = [e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_n]$, 则可以得到下面的结论.

推论 5.15 设 $A \in \mathbb{R}^{n\times n}$ 是对称矩阵, $\tilde{A}$ 是 A 的一个 $n-1$ 阶主子矩阵, A 和 $\tilde{A}$ 的特征值分别为

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \quad \text{和} \quad \tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_{n-1},$$

则有

$$\lambda_1 \geq \tilde{\lambda}_1 \geq \lambda_2 \geq \tilde{\lambda}_2 \cdots \geq \tilde{\lambda}_{n-1} \geq \lambda_n.$$

反复应用上面的推论, 即可得到下面的结论.

推论 5.16 设 $A \in \mathbb{R}^{n\times n}$ 是对称矩阵, $\tilde{A}$ 是 A 的一个 k 阶主子矩阵 $(1 \leq k \leq n-1)$, A 和 $\tilde{A}$ 的特征值分别为

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \quad \text{和} \quad \tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_k,$$

则有

$$\lambda_i \geq \tilde{\lambda}_i \geq \lambda_{n-k+i}, \quad i = 1, 2, \dots, k.$$

5.8.2 对称矩阵特征值的扰动分析

设 $A \in \mathbb{R}^{n\times n}$ 是对称矩阵, 扰动矩阵 $E \in \mathbb{R}^{n\times n}$ 也是对称矩阵, 下面讨论 $A+E$ 的特征值与 A 的特征值之间的关系.

由极小极大定理, 我们可以证明下面的性质.

定理 5.17 设 $A \in \mathbb{R}^{n\times n}$ 和 $B = A + E \in \mathbb{R}^{n\times n}$ 都是对称矩阵, 其特征值分别为

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \quad \text{和} \quad \tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_n.$$

假定 E 的最大和最小特征值分别为 μ_1 和 μ_n, 则有

$$\lambda_i + \mu_1 \geq \tilde{\lambda}_i \geq \lambda_i + \mu_n, \quad i = 1, 2, \dots, n.$$

证明. 由 Courant-Fischer 定理 5.12 和 Rayleigh-Ritz 定理 5.13 可知

$$\begin{aligned}
\tilde{\lambda}_i &= \min_{\mathbb{V}\in\mathbb{S}^n_{n-i+1}} \max_{x\in\mathbb{V},\, x\neq 0} \frac{x^TBx}{x^Tx} \\
&= \min_{\mathbb{V}\in\mathbb{S}^n_{n-i+1}} \max_{x\in\mathbb{V},\, x\neq 0} \left(\frac{x^TAx}{x^Tx} + \frac{x^TEx}{x^Tx}\right) \\
&\leq \min_{\mathbb{V}\in\mathbb{S}^n_{n-i+1}} \max_{x\in\mathbb{V},\, x\neq 0} \left(\frac{x^TAx}{x^Tx} + \mu_1\right) \\
&= \min_{\mathbb{V}\in\mathbb{S}^n_{n-i+1}} \max_{x\in\mathbb{V},\, x\neq 0} \frac{x^TAx}{x^Tx} + \mu_1
\end{aligned}$$

$$= \lambda_i + \mu_1.$$

同理可得

$$\begin{aligned}
\tilde{\lambda}_i &= \max_{\mathbb{U}\in\mathbb{S}_i^n} \min_{x\in\mathbb{U},\, x\neq 0} \frac{x^TBx}{x^Tx} \\
&= \max_{\mathbb{U}\in\mathbb{S}_i^n} \min_{x\in\mathbb{U},\, x\neq 0} \left(\frac{x^TAx}{x^Tx} + \frac{x^TEx}{x^Tx}\right) \\
&\geq \max_{\mathbb{U}\in\mathbb{S}_i^n} \min_{x\in\mathbb{U},\, x\neq 0} \left(\frac{x^TAx}{x^Tx} + \mu_n\right) \\
&= \max_{\mathbb{U}\in\mathbb{S}_i^n} \min_{x\in\mathbb{U},\, x\neq 0} \frac{x^TAx}{x^Tx} + \mu_n \\
&= \lambda_i + \mu_n.
\end{aligned}$$

所以定理结论成立. □

根据这个定理, 我们立即可以得到下面的 Weyl 定理.

定理 5.18 (Weyl) 设 $A \in \mathbb{R}^{n\times n}$ 和 $B = A + E \in \mathbb{R}^{n\times n}$ 都是对称矩阵, 其特征值分别为 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ 和 $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_n$, 则

$$|\tilde{\lambda}_j - \lambda_j| \leq \|E\|_2, \quad j = 1, 2, \ldots, n.$$

该定理的结论可以推广到奇异值情形. 我们首先给出下面的引理.

引理 5.7 设 $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ 的奇异值分解为 $A = U\Sigma V$, 其中 $U = [u_1, \ldots, u_n] \in \mathbb{R}^{m\times n}$ 为列正交矩阵, $V = [v_1, \ldots, v_n] \in \mathbb{R}^{n\times n}$ 为正交矩阵, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$. 将 U 扩展成 $n \times n$ 的正交矩阵 $[U, \tilde{U}] = [u_1, \ldots, u_n, \tilde{u}_1, \ldots, \tilde{u}_{m-n}]$, 令

$$H = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix} \in \mathbb{R}^{(m+n)\times(m+n)},$$

则 H 对称, 且特征值为 $\pm\sigma_i$ 和 0 (其中 0 至少为 $m-n$ 重特征值), 对应的特征向量分别为 $\dfrac{\sqrt{2}}{2}\begin{bmatrix} v_i \\ \pm u_i \end{bmatrix}$, $i = 1, 2, \ldots, n$, 和 $\begin{bmatrix} 0 \\ \tilde{u}_j \end{bmatrix}$, $j = 1, 2, \ldots, m-n$.

证明. 直接代入验证即可. □

由上面的引理和 Weyl 定理 5.18 立即可得

定理 5.19 设 $A, B \in \mathbb{R}^{m\times n}$ $(m \geq n)$, 它们的奇异值分别为 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ 和 $\tilde{\sigma}_1 \geq \tilde{\sigma}_2 \geq \cdots \geq \tilde{\sigma}_n$. 则

$$|\tilde{\sigma}_j - \sigma_j| \leq \|B - A\|_2, \quad j = 1, 2, \ldots, n.$$

5.8.3 对称矩阵特征向量的扰动

定义 5.3 设 $A \in \mathbb{R}^{n\times n}$ 的特征值为 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, 则 λ_i 与其余特征值之间的**间隙** (gap) 定义为

$$\mathrm{gap}(\lambda_i, A) = \min_{j\neq i} |\lambda_j - \lambda_i|.$$

有时简记为 $\mathrm{gap}(\lambda_i)$.

☞ 特征向量的敏感性依赖于其对应的特征值的 gap, 一般来说, gap 越小, 特征向量越敏感.

例 5.2 设

$$A = \begin{bmatrix} 1+g & \\ & 1 \end{bmatrix}, \quad E = \begin{bmatrix} 0 & \varepsilon \\ \varepsilon & 0 \end{bmatrix}, \quad (0 < \varepsilon < g)$$

则 A 的特征值为 $\lambda_1 = 1+g, \lambda_2 = 1$, 对应的单位特征向量为 $q_1 = e_1, q_2 = e_2$. $A+E$ 的特征值为 $\hat{\lambda}_{1,2} = 1 + (g \pm \sqrt{g^2+4\varepsilon^2})/2$, 对应的单位特征向量为

$$\begin{aligned}
\hat{q}_1 = \beta_1 \cdot \begin{bmatrix} 1 \\ \dfrac{\sqrt{1+4\varepsilon^2/g^2}-1}{2\varepsilon/g} \end{bmatrix} &= \beta_1 \cdot \begin{bmatrix} 1 \\ \dfrac{\sqrt{(1+2\varepsilon^2/g^2)^2 - 4(\varepsilon/g)^4}-1}{2\varepsilon/g} \end{bmatrix} \\
&\approx \beta_1 \cdot \begin{bmatrix} 1 \\ \dfrac{(1+2\varepsilon^2/g^2)-1}{2\varepsilon/g} \end{bmatrix} \\
&= \frac{1}{\sqrt{1+\varepsilon^2/g^2}} \begin{bmatrix} 1 \\ \varepsilon/g \end{bmatrix}, \\
\hat{q}_2 = \beta_2 \cdot \begin{bmatrix} 1 \\ \dfrac{-\sqrt{1+4\varepsilon^2/g^2}-1}{2\varepsilon/g} \end{bmatrix} &\approx \frac{1}{\sqrt{1+\varepsilon^2/g^2}} \begin{bmatrix} -\varepsilon/g \\ 1 \end{bmatrix},
\end{aligned}$$

其中 β_1, β_2 为规范化因子. 故特征向量的扰动约为 ε/g, 与特征值的间隙 $\mathrm{gap}(\lambda_i, A) = g$ 成反比.

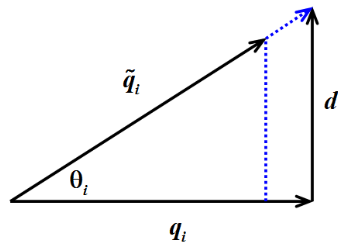
定理 5.20 设 $A = Q\Lambda Q^T$ 和 $A+E = \tilde{Q}\tilde{\Lambda}\tilde{Q}^T$ 分别为对称矩阵 $A \in \mathbb{R}^{n\times n}$ 和 $A+E \in \mathbb{R}^{n\times n}$ 的特征值分解, 其中 $Q = [q_1, q_2, \ldots, q_n]$ 和 $\tilde{Q} = [\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_n]$ 均为正交矩阵, 且 $\tilde{q}_i$ 为 q_i 对应的扰动特征向量. 用 θ_i 表示 q_i 和 $\tilde{q}_i$ 之间的锐角, 则当 $\mathrm{gap}(\lambda_i, A) > 0$ 时

$$\frac{1}{2}\sin 2\theta_i \leq \frac{\|E\|_2}{\mathrm{gap}(\lambda_i, A)}.$$

类似地, 当 $\mathrm{gap}(\tilde{\lambda}_i, A+E) > 0$ 时

$$\frac{1}{2}\sin 2\theta_i \leq \frac{\|E\|_2}{\mathrm{gap}(\tilde{\lambda}_i, A+E)}.$$

证明. 如右图所示, 令 $d = \tilde{q}_i/\cos\theta_i - q_i$, 即 $\tilde{q}_i = (q_i + d)\cos\theta_i$. 则



$$d^Tq_i = 0, \quad \tan\theta_i = \|d\|_2, \quad \sec\theta_i = \|q_i + d\|_2.$$

令 $\eta = \tilde{\lambda}_i - \lambda_i$, 由 $(A+E)\tilde{q}_i = \tilde{\lambda}_i\tilde{q}_i$ 可得

$$(A+E)(q_i+d) = (\eta+\lambda_i)(q_i+d),$$

将 $Aq_i = \lambda_i q_i$ 代入后整理可得

$$(\eta I - E)(q_i + d) = (A - \lambda_i I)d.$$

又 $q_i^T(A-\lambda_i I) = \big((A-\lambda_i I)q_i\big)^T = 0$, 故用 q_i^T 左乘上式两边可得

$$q_i^T(\eta I - E)(q_i+d) = q_i^T(A-\lambda_i I)d = 0, \tag{5.12}$$

即 $(\eta I - E)(q_i+d) \in \mathrm{span}\{q_i\}^{\perp} = \mathrm{span}\{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_n\}$. 所以可设 $(\eta I - E)(q_i+d) = \sum_{j\neq i}\alpha_j q_j$. 又 $q_i^T d = 0$, 故可设 $d = \sum_{j\neq i}\delta_j q_j$. 所以

$$\begin{aligned}
\sum_{j\neq i}\alpha_j q_j &= (\eta I - E)(q_i+d) = (A-\lambda_i I)d \\
&= (A-\lambda_i I)\sum_{j\neq i}\delta_j q_j = \sum_{j\neq i}\delta_j(A-\lambda_i I)q_j = \sum_{j\neq i}\delta_j(\lambda_j - \lambda_i)q_j.
\end{aligned}$$

由于 $q_1, q_2, \dots, q_n$ 线性无关, 故可得 $\delta_j(\lambda_j - \lambda_i) = \alpha_j$. 又 $\mathrm{gap}(\lambda_i, A) > 0$, 即 $j \neq i$ 时 $\lambda_j \neq \lambda_i$, 所以 $\delta_i = \frac{\alpha_i}{\lambda_j - \lambda_i}$, 因此

$$d = \sum_{j\neq i}\frac{\alpha_j}{\lambda_j - \lambda_i}q_j.$$

注意到 $q_i^T d = 0$ 且 $q_i^T q_i = 1$, 所以由 (5.12) 可得

$$\eta = q_i^T E(q_i + d).$$

故

$$\begin{aligned}
(\eta I - E)(q_i+d) &= (q_i+d)\eta - E(q_i+d) \\
&= (q_i+d)q_i^T E(q_i+d) - E(q_i+d) \\
&= \big((q_i+d)q_i^T - I\big)E(q_i+d).
\end{aligned}$$

由习题 5.4 可知 $\|(q_i+d)q_i^T - I\|_2 = \|q_i+d\|_2$, 故

$$\begin{aligned}
\tan\theta_i = \|d\|_2 &= \left(\sum_{j\neq i}\left|\frac{\alpha_j}{\lambda_j - \lambda_i}\right|^2\right)^{1/2} \\
&\leq \left(\sum_{j\neq i}\left|\frac{\alpha_j}{\mathrm{gap}(\lambda_i, A)}\right|^2\right)^{1/2} \\
&= \frac{1}{\mathrm{gap}(\lambda_i, A)}\left(\sum_{j\neq i}\alpha_j^2\right)^{1/2}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\mathrm{gap}(\lambda_i, A)} \left\| \sum_{j\neq i} \alpha_j q_j \right\|_2 \\
&= \frac{1}{\mathrm{gap}(\lambda_i, A)} \|(\eta I - E)(q_i + d)\|_2 \\
&\leq \frac{1}{\mathrm{gap}(\lambda_i, A)} \|(q_i + d)q_i^T - I\|_2 \cdot \|E\|_2 \cdot \|(q_i + d)\|_2 \\
&= \frac{1}{\mathrm{gap}(\lambda_i, A)} \|(q_i + d)\|_2^2 \cdot \|E\|_2 \\
&= \frac{1}{\mathrm{gap}(\lambda_i, A)} \cdot \frac{1}{\cos^2\theta_i} \|E\|_2.
\end{aligned}
$$

即

$$
\frac{1}{2}\sin 2\theta_i = \sin\theta_i \cos\theta_i = \tan\theta_i \cos\theta_i^2 \leq \frac{1}{\mathrm{gap}(\lambda_i, A)} \|E\|_2.
$$

将 $A+E$ 看作原矩阵, $(A+E)-E$ 看作是扰动矩阵, 则可证明第二个结论. □

- 当 $\theta_i \ll 1$ 时, $\frac{1}{2}\sin 2\theta_i \approx \theta_i \approx \sin\theta_i$;
- 当 $\|E\|_2 \geq \frac{1}{2}\mathrm{gap}(\lambda_i, A)$ 时, 定理中给出的上界就失去了实际意义;
- 在该定理中, 没有对特征值进行排序;
- 在实际计算中, 我们通常所知道的是 $\mathrm{gap}(\tilde{\lambda}_i, A+E)$.

5.8.4 Rayleigh 商逼近

定理 5.21 设对称矩阵 $A \in \mathbb{R}^{n\times n}$ 的特征值为 $\lambda_1, \lambda_2, \ldots, \lambda_n$,

(1) 若 $x \in \mathbb{R}^n$ 是单位向量, $\beta \in \mathbb{R}$, 则

$$
\min_{1\leq i\leq n} |\lambda_i - \beta| \leq \|Ax - \beta x\|_2; \tag{5.13}
$$

(2) 对于给定的非零向量 $x \in \mathbb{R}^n$, 当 $\beta = \rho(x)$ 时, $\|Ax - \beta x\|_2$ 达到最小, 即

$$
\min_{\beta\in\mathbb{R}} \|Ax - \beta x\|_2 = \|Ax - \rho(x)x\|_2; \tag{5.14}
$$

(3) 令 $r = Ax - \rho(x)x$, 设 λ_i 是距离 $\rho(x)$ 最近的特征值, $\mathrm{gap}' = \min_{j\neq i} |\lambda_j - \rho(x)|$, θ 是 x 和 q_i 之间的锐角, 其中 q_i 是 λ_i 对应的单位特征向量, 则

$$
\sin\theta \leq \frac{\|r\|_2}{\mathrm{gap}'} \quad \text{且} \quad |\lambda_i - \rho(x)| \leq \frac{\|r\|_2^2}{\mathrm{gap}'}. \tag{5.15}
$$

证明. (1) 若 β 是 A 的特征值, 则结论显然成立.

若 β 不是 A 的特征值, 则 $A - \beta I$ 非奇异, 故

$$
1 = \|x\|_2 = \|(A-\beta I)^{-1}(A-\beta I)x\|_2 \leq \|(A-\beta I)^{-1}\|_2 \cdot \|(A-\beta I)x\|_2. \tag{5.16}
$$

由于 $A - \beta I$ 对称, 且特征值为 $\lambda_i - \beta$, 故

$$
\|(A-\beta I)^{-1}\|_2 = \frac{1}{\min_{1\leq i\leq n} |\lambda_i - \beta|}.
$$

代入 (5.16) 即可知结论成立.

(2) 由于

$$x^T\big(Ax-\rho(x)x\big)=x^TAx-\frac{x^TAx}{x^Tx}\,x^Tx=0,$$

即 $x\perp(Ax-\rho(x)x)$. 所以

$$\begin{aligned}\|Ax-\beta x\|_2^2&=\big\|\big(A-\rho(x)\big)x+\big(\rho(x)-\beta\big)x\big\|_2^2\\&=\|Ax-\rho(x)x\|_2^2+\big\|\big(\rho(x)-\beta\big)x\big\|_2^2\\&\geq\|Ax-\rho(x)x\|_2^2,\end{aligned}$$

所以当 $\beta=\rho(x)$ 时, $\|Ax-\beta x\|_2$ 达到最小.

(3) 略 □

☞ 由 (5.13) 可知, 在幂迭代和反迭代中可以使用残量 $\|Ax-\tilde{\lambda}x\|_2<tol$ 作为停机准则, 这里 $\tilde{\lambda}$ 是迭代过程中计算得到的近似特征值. 等式 (5.14) 则解释了为什么用 Rayleigh 商来近似特征值.

☞ 不等式 (5.15) 表明 $|\lambda_i-\rho(x)|$ 的值与残量范数 $\|r\|_2$ 的平方成正比, 这个结论是 Rayleigh 商迭代局部三次收敛的基础.

5.8.5 相对扰动分析

这里主要讨论 A 和 X^TAX 的特征值和特征向量之间的扰动关系, 其中 X 非奇异且满足 $\|X^TX-I\|_2=\varepsilon$. 这是因为在计算特征向量时, 由于舍入误差的原因, 最后得到的正交矩阵 Q 会带有误差, 从而失去正交性.

定理 5.22 (相对 Weyl 定理) 设对称矩阵 A 和 X^TAX 的特征值分别为 $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_n$ 和 $\tilde{\lambda}_1\geq\tilde{\lambda}_2\geq\cdots\geq\tilde{\lambda}_n$, 令 $\varepsilon=\|X^TX-I\|_2$, 则

$$|\tilde{\lambda}_i-\lambda_i|\leq\varepsilon|\lambda_i|\quad \text{或}\quad \frac{|\tilde{\lambda}_i-\lambda_i|}{|\lambda_i|}\leq\varepsilon\quad (\textit{if}\ \lambda_i\neq 0).$$

证明. 因为 $A-\lambda_iI$ 的第 i 个特征值为 0, 故由 Sylvester 惯性定理 5.6 可知

$$X^T(A-\lambda_iI)X=(X^TAX-\lambda_iI)+\lambda_i(I-X^TX)$$

的第 i 个特征值也为 0. 由 Weyl 定理 5.18 可知

$$|(\tilde{\lambda}_i-\lambda_i)-0\|\leq\|\lambda_i(I-X^TX)\|_2=\varepsilon|\lambda_i|,$$

即定理结论成立 □

☞ 当 X 正交时, $\varepsilon=0$, 故 X^TAX 与 A 有相同的特征值. 当 X 几乎正交时, ε 很小, 此时 X^TAX 与 A 的特征值几乎相同.

推论 5.23 设 G 和 Y^TGX 的奇异值分别为 $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_n$ 和 $\tilde{\sigma}_1 \ge \tilde{\sigma}_2 \ge \cdots \ge \tilde{\sigma}_n$, 令 $\varepsilon = \max\left\{\|X^TX - I\|_2, \|Y^TY - I\|_2\right\}$, 则

$$|\tilde{\sigma}_i - \sigma_i| \le \varepsilon|\sigma_i| \quad \text{或} \quad \frac{|\tilde{\sigma}_i - \sigma_i|}{|\sigma_i|} \le \varepsilon \quad (\textit{if}\ \sigma_i \neq 0).$$

下面给出特征向量的相对扰动性质.

定义 5.4 设 $A \in \mathbb{R}^{n\times n}$ 的特征值为 $\lambda_1, \lambda_2, \ldots, \lambda_n$, 若 $\lambda_i \neq 0$, 则 λ_i 与其余特征值之间的**相对间隙 (relative gap)** 定义为

$$\mathrm{relgap}(\lambda_i, A) = \min_{j\neq i} \frac{|\lambda_j - \lambda_i|}{|\lambda_i|}.$$

定理 5.24 设 $A \in \mathbb{R}^{n\times n}$ 和 $X^TAX \in \mathbb{R}^{n\times n}$ 的特征值分解分别为 $A = Q\Lambda Q^T$ 和 $X^TAX = \tilde{Q}\tilde{\Lambda}\tilde{Q}^T$, 其中 $Q = [q_1, q_2, \ldots, q_n]$ 和 $\tilde{Q} = [\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_n]$ 均为正交矩阵, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, $\tilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_n)$ 且 $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$, $\tilde{\lambda}_1 \ge \tilde{\lambda}_2 \ge \cdots \ge \tilde{\lambda}_n$. 设 θ_i 表示 q_i 和 $\tilde{q}_i$ 之间的锐角, 令 $\varepsilon_1 = \|I - X^{-T}X^{-1}\|_2$, $\varepsilon_2 = \|X - I\|_2$, 若 $\varepsilon_1 < 1$ 且 $\mathrm{relgap}(\tilde{\lambda}_i, X^TAX) > 0$, 则

$$\frac{1}{2}\sin 2\theta_i \le \frac{\varepsilon_1}{1-\varepsilon_1} \cdot \frac{1}{\mathrm{relgap}(\tilde{\lambda}_i, X^TAX)} + \varepsilon_2.$$

证明. 设 $\eta = \tilde{\lambda}_i - \lambda_i$, $H = A - \tilde{\lambda}_i I$, $F = \tilde{\lambda}_i(I - X^{-T}X^{-1})$, 则 $Hq_i = Aq_i - \tilde{\lambda}_i q_i = -\eta q_i$,

$$H + F = A - \tilde{\lambda}_i X^{-T}X^{-1} = X^{-T}(X^TAX - \tilde{\lambda}_i I)X^{-1}.$$

故 $(H+F)(X\tilde{q}_i) = 0$. 即 $X\tilde{q}_i$ 是 $H+F$ 的第 i 个特征值 $\hat{\lambda}_i = 0$ 的一个特征向量. 设 θ_1 是 q_i 与 $X\tilde{q}_i$ 之间的锐角, 由定理 5.20 可知

$$\frac{1}{2}\sin 2\theta_1 \le \frac{\|F\|_2}{\mathrm{gap}(\lambda_i, H+F)} = \frac{\varepsilon_1|\tilde{\lambda}_i|}{\mathrm{gap}(\lambda_i, H+F)}. \tag{5.17}$$

由于 $\hat{\lambda}_i = 0$, 故 $\mathrm{gap}(\lambda_i, H+F)$ 即为 $H+F$ 的最小非零特征值的绝对值. 又 $X^T(H+F)X = X^TAX - \tilde{\lambda}_i I$ 的特征值为 $\tilde{\lambda}_j - \tilde{\lambda}_i$, $j = 1, 2, \ldots, n$, 且

$$\tilde{\lambda}_1 - \tilde{\lambda}_i \ge \tilde{\lambda}_2 - \tilde{\lambda}_i \ge \cdots \ge \tilde{\lambda}_n - \tilde{\lambda}_i,$$

所以由相对 Weyl 定理 5.22 可知

$$|\hat{\lambda}_j - (\tilde{\lambda}_j - \tilde{\lambda}_i)| \le \varepsilon_1|\tilde{\lambda}_j - \tilde{\lambda}_i|.$$

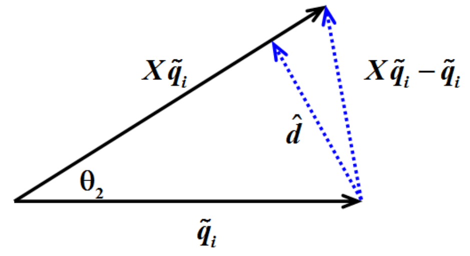
这里 $\hat{\lambda}_j$ 表示 $H+F$ 的第 j 个特征值 (按降序排列). 因此 $|\hat{\lambda}_j| \ge (1-\varepsilon_1)|\tilde{\lambda}_j - \tilde{\lambda}_i|$, 故

$$\mathrm{gap}(\hat{\lambda}_i, H+F) \ge (1-\varepsilon_1)\mathrm{gap}(\tilde{\lambda}_i, X^TAX).$$

代入 (5.17) 可得

$$\frac{1}{2}\sin 2\theta_1 \le \frac{\varepsilon_1|\tilde{\lambda}_i|}{(1-\varepsilon_1)\mathrm{gap}(\tilde{\lambda}_i, X^TAX)} = \frac{\varepsilon_1}{1-\varepsilon_1} \cdot \frac{1}{\mathrm{relgap}(\tilde{\lambda}_i, X^TAX)}.$$

设 θ_2 是 $X\tilde{q}_i$ 与 $\tilde{q}_i$ 之间的锐角, 则由右图可知



$$\begin{aligned}\sin\theta_2 = \|\tilde{q}_i\|_2 \sin\theta_2 &\le \|X\tilde{q}_i - \tilde{q}_i\|_2 \\ &\le \|X - I\|_2 \cdot \|\tilde{q}_i\|_2 \\ &= \varepsilon_2.\end{aligned}$$

又 $\theta_i \le \theta_1 + \theta_2$, 故

$$\begin{aligned}\frac{1}{2}\sin 2\theta_i &\le \frac{1}{2}\sin 2\theta_1 + \frac{1}{2}\sin 2\theta_2 \\ &\le \frac{1}{2}\sin 2\theta_1 + \sin\theta_2 \\ &\le \frac{\varepsilon_1}{1-\varepsilon_1} \cdot \frac{1}{\operatorname{relgap}(\tilde{\lambda}_i, X^T A X)} + \varepsilon_2,\end{aligned}$$

即定理结论成立. □

5.9 课后习题

5.1 设 $\lambda \in \mathbb{R}$ 是对称矩阵 $A \in \mathbb{R}^{n\times n}$ 的一个特征值, 对应的特征向量为 $x \in \mathbb{R}^n$. 若 $\tilde{x} \in \mathbb{R}^n$ 是 x 的一个 $O(\varepsilon)$ 近似, 即 $\tilde{x} = x + O(\varepsilon)$, 证明:

$$\frac{\tilde{x}^T A\tilde{x}}{\tilde{x}^T\tilde{x}} = \lambda + O(\varepsilon^2),$$

即 $\tilde{x}$ 对应的 Rayleigh 商是 λ 的 $O(\varepsilon^2)$ 逼近.

5.2 设 $A = D + \alpha uu^T$, 其中 $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$, $u = [u_1, u_2, \ldots, u_n]^T$. 证明:
(1) 若 $d_i = d_{i+1}$ 或 $u_i = 0$, 则 d_i 是 A 的特征值;
(2) 若 $u_i = 0$, 则 e_i 是与 d_i 对应的特征向量;
(3) 若 $d_i = d_{i+1}$, 试给出与 d_i 对应的特征向量.

5.3 设 $x, y \in \mathbb{R}^n$, 试证明: $\det(I + xy^T) = 1 + y^T x$. (注: 在复数域也成立)

5.4 设 $q \in \mathbb{R}^n$ 满足 $\|q\|_2 = 1$. 对任意与 q 正交的向量 $d \in \mathbb{R}^n$, 试证明:

$$\|(q + d)q^T - I\|_2 = \|q + d\|_2.$$

5.5 设 $A \in \mathbb{C}^{n\times n}$ 是 skew-Hermitian 矩阵, 即 $A^* = -A$. 证明:
(1) A 的非零特征值是纯虚数;
(2) $I + A$ 非奇异;
(3) 矩阵 $(I + A)^{-1}(I - A)$ 是酉矩阵. (该矩阵称为 A 的 Cayley 变换)

5.6 设 $B \in \mathbb{R}^{m\times n}, m \geq n$ 且 $\|B\|_2 < 1$. 若 $A = \begin{bmatrix} I & B \\ B^T & I \end{bmatrix}$, 证明:

$$\kappa_2(A) = \frac{1 + \|B\|_2}{1 - \|B\|_2}.$$

5.7 设 $x \in \mathbb{R}^n$ 是一个正向量, 即 $x_i > 0$. 证明: 由 x 定义的 Cauchy 矩阵 A

$$a_{ij} = \frac{1}{x_i + x_j}$$

是对称半正定的. (参见詹兴致教授的 “矩阵论”)

思考题

5.8 设 $x, y \in \mathbb{R}^n$, 若 $y^T x$ 只有零特征值, 证明: xy^T 也只有零特征值.
设 $X, Y \in \mathbb{R}^{n\times 2}$, 若 $Y^T X$ 只有零特征值, 则 XY^T 是否也只有零特征值?

实践题

5.9 写出对称矩阵三对角化的完整算法.

5.10 写出带 Wilkinson 位移的计算对称三对角矩阵的特征值和特征向量的完整 QR 算法, 并上机编程实现 (MATLAB).

第六章　线性方程组的迭代解法

求解线性方程组的方法有:

- 直接法
- 经典迭代法: Jacobi/Gauss-Seidel, SOR, AOR
- Krylov 子空间迭代法
- 多重网格 (Multigrid)
- 快速多极子算法 (Fast multipole)

每种方法通常只是对某些类方程有效. 在实际应用中, 这些方法可以结合使用, 如混合 (hybrid) 算法, 预处理算法 (preconditioning). 本章以 Poisson 方程为例, 描述两类基本的线性方程组迭代算法:

- 经典迭代算法: Jacobi, GS, SOR, AOR, SSOR
- Krylov 子空间迭代算法: CG, GMRES
- 快速 Fourier 变换 (FFT)
- 多重网格 (Multigrid)

更多迭代算法可参见 [3].

6.1 离散 Poisson 方程

6.1.1 一维 Poisson 方程

考虑带 Dirichlet 边界条件的一维 Poisson 方程

$$\begin{cases} -\dfrac{d^2u(x)}{dx^2} = f(x), \quad 0 < x < 1, \\ u(0) = a, u(1) = b, \end{cases} \tag{6.1}$$

其中 $f(x)$ 是给定的函数, $u(x)$ 是需要计算的函数.

差分离散

取步长 $h = \dfrac{1}{n+1}$, 节点 $x_i = ih, i = 0, 1, 2, \dots, n+1$. 采用中心差分离散, 即

$$-\left.\frac{d^2u(x)}{dx^2}\right|_{x_i} = \frac{2u(x_i) - u(x_{i-1}) - u(x_{i+1})}{h^2} + O\left(h^2 \cdot \left\|\frac{d^4u}{dx^4}\right\|_\infty\right), \quad i = 1, 2, \dots, n.$$

代入一维 poisson 方程 (6.1), 舍去高阶项后可得 Poisson 方程在 x_i 点的近似方程

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i,$$

其中 $f_i = f(x_i)$, u_i 为 $u(x_i)$ 的近似. 令 $i = 1, 2, \dots, n$, 则可得 n 个线性方程, 写成矩阵形式

$$T_n u = f, \tag{6.2}$$

其中

$$T_n = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \triangleq \text{tridiag}(-1, 2, -1), \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}, \quad f = \begin{bmatrix} f_1 + u_0 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n + u_{n+1} \end{bmatrix}. \tag{6.3}$$

系数矩阵 T_n 的性质

引理 6.1 T_n 的特征值和对应的特征向量分别为

$$\lambda_k = 2 - 2\cos\frac{k\pi}{n+1},$$
$$z_k = \sqrt{\frac{2}{n+1}} \cdot \left[\sin\frac{k\pi}{n+1}, \sin\frac{2k\pi}{n+1}, \dots, \sin\frac{nk\pi}{n+1}\right]^T, \quad k = 1, 2, \dots, n,$$

即 $T_n = Z\Lambda Z^T$, 其中 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, $Z = [z_1, z_2, \dots, z_n]$.

证明. 直接代入验证即可. □

引理 6.2 更一般地, 设 $T = \text{tridiag}(a, b, c) \in \mathbb{R}^{n\times n}$, 则 T 的特征值为

$$\lambda_k = b - 2\sqrt{ac}\cos\frac{k\pi}{n+1}, \quad k = 1, 2, \dots, n,$$

对应的特征向量为 z_k, 其第 j 个分量为

$$z_k(j) = \left(\frac{a}{c}\right)^{\frac{j}{2}} \sin\frac{jk\pi}{n+1}.$$

特别地, 若 $a = c = 1$, 则对应的单位特征向量为

$$z_k = \sqrt{\frac{2}{n+1}} \cdot \left[\sin\frac{k\pi}{n+1}, \sin\frac{2k\pi}{n+1}, \dots, \sin\frac{nk\pi}{n+1}\right]^T.$$

由引理 6.1 可知, T_n 是对称正定的, 其最大特征值为

$$2\left(1 - \cos\frac{n\pi}{n+1}\right) = 4\sin^2\frac{n\pi}{2(n+1)} \approx 4,$$

最小特征值为

$$2\left(1 - \cos\frac{\pi}{n+1}\right) = 4\sin^2\frac{\pi}{2(n+1)} \approx \left(\frac{\pi}{n+1}\right)^2.$$

因此, 当 n 很大时, T_n 的谱条件数约为

$$\kappa_2(T_n) \approx \frac{4(n+1)^2}{\pi^2}.$$

6.1.2 二维 Poisson 方程

现在考虑二维 Poisson 方程

$$\begin{cases} -\Delta u(x,y) = -\dfrac{\partial^2 u(x,y)}{\partial x^2} - \dfrac{\partial^2 u(x,y)}{\partial y^2} = f(x,y), & (x,y) \in \Omega, \\ u(x,y) = v_0(x,y), \quad (x,y) \in \partial\Omega \end{cases} \tag{6.4}$$

其中 $\Omega = [0,1] \times [0,1]$ 为求解区域, $\partial\Omega$ 表示 Ω 的边界.

五点差分离散

为了简单起见, 我们在 x-方向和 y-方向取相同的步长 $h = \dfrac{1}{n+1}$, 节点 $x_i = ih$, $y_j = jh$, $i, j = 0, 1, 2, \dots, n$. 在 x-方向和 y-方向同时采用中心差分离散可得

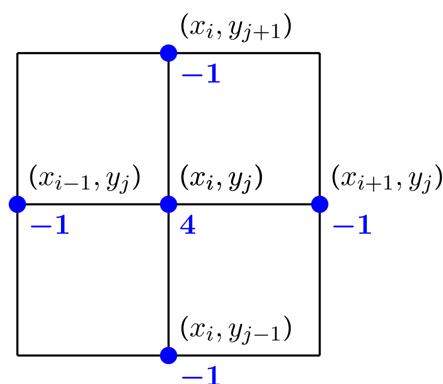
$$\left.\frac{\partial^2 u(x,y)}{\partial x^2}\right|_{(x_i,y_j)} \approx \frac{2u(x_i,y_j) - u(x_{i-1},y_j) - u(x_{i+1},y_j)}{h^2}$$

$$\left.\frac{\partial^2 u(x,y)}{\partial y^2}\right|_{(x_i,y_j)} \approx \frac{2u(x_i,y_j) - u(x_i,y_{j-1}) - u(x_i,y_{j+1})}{h^2}.$$

代入二维 Poisson 方程 (6.4) 即得 Poisson 方程在 (x_i, y_j) 点的近似离散方程

$$4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = h^2 f_{i,j},$$

其中 $f_{ij} = f(x_i, y_j)$, $u_{i,j}$ 为 $u(x_i, y_j)$ 的近似.



写成矩阵形式即为

$$Tu = h^2 f, \tag{6.5}$$

其中 $T \triangleq I \otimes T_n + T_n \otimes I$, $u = [u_{1,1}, \dots, u_{n,1}, u_{1,2}, \dots, u_{n,2}, \dots, u_{1,n}, \dots, u_{n,n}]$,. 这里 $\otimes$ 表示 Kronecker 乘积, T_n 为一维 Poisson 方程离散后的系数矩阵, 即 (6.3).

在后面介绍的算法时, 我们都以二维离散 Poisson 方程 (6.5) 为例.

系数矩阵 T 的性质

因为 $T = I \otimes T_n + T_n \otimes I$, 由 Kronecker 乘积的性质即得

定理 6.1 设 $T_n = Z\Lambda Z^T$, 其中 $Z = [z_1, z_2, \dots, z_n]$ 为正交阵, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ 为对角阵, 则 T 的特征值分解为

$$T = (Z \otimes Z)(I \otimes \Lambda + \Lambda \otimes I)(Z \otimes Z)^T,$$

即 T 的特征值为 $\lambda_i + \lambda_j$, 对应的特征向量为 $z_i \otimes z_j$ $(i, j = 1, 2, \dots, n)$.

由于 T 对称正定, 其条件数为

$$\kappa(T) = \frac{\lambda_{max}(T)}{\lambda_{min}(T)} = \frac{1 - \cos\dfrac{n\pi}{n+1}}{1 - \cos\dfrac{\pi}{n+1}} = \frac{\sin^2\dfrac{n\pi}{2(n+1)}}{\sin^2\dfrac{\pi}{2(n+1)}} \approx \frac{4(n+1)^2}{\pi^2}.$$

故当 n 越来越大时, $\kappa(T) \to \infty$, 即 T 越来越病态.

类似地, 三维 poisson 方程采用中心差分离散后的系数矩阵为

$$T_n \otimes I \otimes I + I \otimes T_n \otimes I + I \otimes I \otimes T_n.$$

二维离散 Poisson 方程的常用算法

假定网格剖分为 $n \times n$, 并记 $n_s = n^2$.

	方法	串行时间	存储空间
直接法	稠密 Cholesky 分解	$\mathcal{O}(n_s^3)$	$\mathcal{O}(n_s^2)$
	显式求逆	$\mathcal{O}(n_s^2)$	$\mathcal{O}(n_s^2)$
	带状 Cholesky 分解	$\mathcal{O}(n_s^2)$	$\mathcal{O}(n_s^{3/2})$
	稀疏 Cholesky 分解	$\mathcal{O}(n_s^{3/2})$	$\mathcal{O}(n_s \log n_s)$
古典迭代	Jacobi	$\mathcal{O}(n_s^2)$	$\mathcal{O}(n_s)$
	Gauss-Seidel	$\mathcal{O}(n_s^2)$	$\mathcal{O}(n_s)$
	SOR	$\mathcal{O}(n_s^{3/2)}$	$\mathcal{O}(n_s)$
	带 Chebyshev 加速的 SSOR	$\mathcal{O}(n_s^{5/4})$	$\mathcal{O}(n_s)$
Krylov 子空间迭代	CG (共轭梯度法)	$\mathcal{O}(n_s^{3/2})$	$\mathcal{O}(n_s)$
	CG (带修正 IC 预处理)	$\mathcal{O}(n_s^{5/4})$	$\mathcal{O}(n_s)$
快速算法	FFT (快速 Fourier 变换)	$\mathcal{O}(n_s \log n_s)$	$\mathcal{O}(n_s)$
	块循环约化	$\mathcal{O}(n_s \log n_s)$	$\mathcal{O}(n_s)$
	Multigrid	$\mathcal{O}(n_s)$	$\mathcal{O}(n_s)$

6.2 古典迭代算法

本节介绍基于矩阵分裂的古典迭代算法 (Classic iterative methods). 内容包括:

- Jacobi 算法
- Gauss-Seidel 算法
- SOR (Successive Over-Relaxation) 算法
- SSOR (Symmetric SOR) 算法
- AOR (Accelerated over-relaxation) 算法

6.2.1 矩阵分裂迭代及收敛性

考虑线性方程组

$$Ax = b, \tag{6.6}$$

其中 $A \in \mathbb{R}^{n\times n}$ 非奇异. 迭代算法的基本思想: 给定一个迭代初始值 $x^{(0)}$, 通过一定的迭代格式生成一个迭代序列 $\{x^{(k)}\}_{k=0}^{\infty}$, 使得

$$\lim_{k\to\infty} x^{(k)} = x_* \triangleq A^{-1}b.$$

定义 6.1 (矩阵分裂 Matrix splitting) 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, 称

$$A = M - N \tag{6.7}$$

为 A 的一个矩阵分裂, 其中 M 非奇异.

给定一个矩阵分裂 (6.7), 则原方程组 (6.6) 就等价于 $Mx = Nx + b$. 于是我们就可以构造出以下的迭代格式

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b \triangleq Gx^{(k)} + g, \quad k = 0, 1, \ldots, \tag{6.8}$$

其中 $G = M^{-1}N$ 称为该迭代格式的迭代矩阵. 取不同的 M, 就可以构造出不同迭代算法.

下面给出向量序列收敛的定义.

定义 6.2 设向量序列 $\{x^{(k)}\}_{k=0}^{\infty} \subset \mathbb{R}^n$. 若存在向量 $x = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$ 使得

$$\lim_{k\to\infty} x_i^{(k)} = x_i, \quad i = 1, 2, \ldots, n,$$

其中 $x_i^{(k)}$ 表示 $x^{(k)}$ 的第 i 个分量. 则称 $x^{(k)}$ 收敛到 x, 即 x 是 $x^{(k)}$ 的极限, 记为

$$\lim_{k\to\infty} x^{(k)} = x.$$

相类似地, 我们可以给出矩阵序列收敛的定义.

定义 6.3 设矩阵序列 $\left\{A^{(k)} = \left[a_{ij}^{(k)}\right]\right\}_{k=0}^{\infty} \subset \mathbb{R}^{n\times n}$. 若存在矩阵 $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ 使得

$$\lim_{k\to\infty} a_{ij}^{(k)} = a_{ij}, \quad i, j = 1, 2, \ldots, n,$$

则称 $A^{(k)}$ 收敛到 A, 即 A 是 $A^{(k)}$ 的极限, 记为

$$\lim_{k\to\infty} A^{(k)} = A.$$

关于向量序列和矩阵序列的收敛性, 我们有下面的结论.

定理 6.2 设向量序列 $\{x^{(k)}\}_{k=0}^{\infty} \subset \mathbb{R}^n$, 矩阵序列 $\left\{A^{(k)} = \left[a_{ij}^{(k)}\right]\right\}_{k=0}^{\infty} \subset \mathbb{R}^{n\times n}$, 则

(1) $\lim\limits_{k\to\infty} x^{(k)} = x \Longleftrightarrow \lim\limits_{k\to\infty} \|x^{(k)} - x\| = 0$, 其中 $\|\cdot\|$ 为任一向量范数;

(2) $\lim\limits_{k\to\infty} A^{(k)} = A \Longleftrightarrow \lim\limits_{k\to\infty} \|A^{(k)} - A\| = 0$, 其中 $\|\cdot\|$ 为任一矩阵范数;

(3) $\lim\limits_{k\to\infty} A^{(k)} = 0 \Longleftrightarrow \lim\limits_{k\to\infty} A^{(k)}x = 0, \forall\, x \in \mathbb{R}^n$.

如果对任意的初始向量 $x^{(0)}$, 都有 $\lim\limits_{k\to\infty} x^{(k)} \to x_*$, 则称迭代格式 (6.8) 是**收敛**的, 否则就称其为**发散**的.

基于矩阵分裂的迭代算法, 其收敛性取决于迭代矩阵的谱半径.

矩阵谱半径及其性质

设 $A \in \mathbb{R}^{n\times n}$, 则称

$$\rho(A) \triangleq \max_{\lambda\in\sigma(A)} |\lambda|$$

为 A 的**谱半径**, 其中 $\sigma(A)$ 表示 A 的所有特征值组成的集合.

谱半径与矩阵范数之间有如下的关系.

引理 6.3 (谱半径与范数的关系) 设 $G \in \mathbb{R}^{n\times n}$, 则

(1) 对任意算子范数, 有 $\rho(G) \leq \|G\|$;

(2) 反之, 对任意 $\varepsilon > 0$, 都存在一个算子范数 $\|\cdot\|_\varepsilon$, 使得 $\|G\|_\varepsilon \leq \rho(G) + \varepsilon$, 其中范数 $\|\cdot\|_\varepsilon$ 依赖于 G 和 ε. 所以, 若 $\rho(G) < 1$, 则存在算子范数 $\|\cdot\|_\varepsilon$, 使得 $\|G\|_\varepsilon < 1$;

证明. (1) 设 λ 是 G 的一个特征值, 对应的特征向量为 $x \neq 0$, 则由 $Gx = \lambda x$ 可得

$$|\lambda| \cdot \|x\| = \|\lambda x\| = \|Gx\| \leq \|G\| \cdot \|x\|.$$

故 $|\lambda| \leq \|G\|$. 所以

$$\rho(G) = \max_{\lambda\in\sigma(G)} |\lambda| \leq \|G\|.$$

(2) 用构造法证明. 设 G 的 Jordan 标准型为 J, 即 $S^{-1}GS = J$. 令 $D = \mathrm{diag}(1, \varepsilon, \varepsilon^2, \ldots, \varepsilon^{n-1})$,

则

$$
(SD)^{-1}G(SD) = D^{-1}JD = \left[\begin{array}{cccc|cccc|ccc}
\lambda_1 & \varepsilon & & & & & & & & & \\
& \ddots & \ddots & & & & & & & & \\
& & \ddots & \varepsilon & & & & & & & \\
& & & \lambda_1 & & & & & & & \\
\hline
& & & & \lambda_2 & \varepsilon & & & & & \\
& & & & & \ddots & \ddots & & & & \\
& & & & & & \ddots & \varepsilon & & & \\
& & & & & & & \lambda_2 & & & \\
\hline
& & & & & & & & \ddots & & \\
& & & & & & & & & \ddots & \\
& & & & & & & & & & \ddots
\end{array}\right].
$$

定义 $\|x\|_\varepsilon \triangleq \|(SD)^{-1}x\|_\infty$. 可以证明 $\|\cdot\|_\varepsilon$ 构成一个向量范数. 由此可得诱导范数

$$
\begin{aligned}
\|G\|_\varepsilon &\triangleq \max_{x\neq 0} \frac{\|Gx\|_\varepsilon}{\|x\|_\varepsilon} \\
&= \max_{x\neq 0} \frac{\|(SD)^{-1}Gx\|_\infty}{\|(SD)^{-1}x\|_\infty} \\
&= \max_{y\neq 0} \frac{\|(SD)^{-1}G(SD)y\|_\infty}{\|y\|_\infty} \\
&= \|(SD)^{-1}G(SD)\|_\infty \\
&\leq \max_{\lambda\in\sigma(G)} \{|\lambda|\} + \varepsilon \\
&= \rho(G) + \varepsilon.
\end{aligned}
$$

□

由定理 6.2 和引理 6.3, 我们可以立即得到下面的结论.

定理 6.3 设矩阵 $G \in \mathbb{R}^{n\times n}$, 则 $\lim\limits_{k\to\infty} G^k = 0$ 当且仅当 $\rho(G) < 1$.

谱半径与算子范数之间的另一个重要性质是

引理 6.4 设 $G \in \mathbb{R}^{n\times n}$, 则对任意算子范数 $\|\cdot\|$, 有

$$
\lim_{k\to\infty} \|G^k\|^{\frac{1}{k}} = \rho(G).
$$

证明. 首先, 我们有

$$
\rho(G)^k = \rho(G^k) \leq \|G^k\|.
$$

另一方面, 对任意 $\varepsilon > 0$, 记

$$
G_\varepsilon = \frac{G}{\rho(G) + \varepsilon}.
$$

则 $\rho(G_\varepsilon) < 1$, 故 $\lim\limits_{k\to\infty} G_\varepsilon^k = 0$. 因此存在正整数 N, 使得当 $k > N$ 时, 有 $\|G_\varepsilon^k\| < 1$, 即

$$\left\| \frac{G^k}{(\rho(G)+\varepsilon)^k} \right\| = \frac{\|G^k\|}{(\rho(G)+\varepsilon)^k} < 1.$$

所以

$$\rho(G) \le \|G^k\|^{\frac{1}{k}} \le \rho(G) + \varepsilon.$$

由 ε 的任意性可知, 当 $k \to \infty$ 时, 有 $\|G^k\|^{\frac{1}{k}} \to \rho(G)$. □

迭代算法收敛性

下面考虑基于矩阵分裂的迭代算法 6.8 的收敛性. 首先给出一个迭代算法收敛的充分条件.

引理 6.5 若存在算子范数 $\|\cdot\|$, 使得 $\|G\| < 1$, 则迭代算法 6.8 收敛.

证明. 由 $x^{(k+1)} = Gx^{(k)} + g$ 和 $x_* = Gx_* + g$ 可得

$$x^{(k+1)} - x_* = G\left(x^{(k)} - x_*\right).$$

故

$$\left\|x^{(k+1)} - x_*\right\| \le \|G\| \cdot \left\|x^{(k)} - x_*\right\|.$$

依此类推, 可得

$$\left\|x^{(k+1)} - x_*\right\| \le \|G\|^{k+1} \cdot \left\|x^{(0)} - x_*\right\|.$$

由于 $\|G\| < 1$, 当 $k \to \infty$ 时, 上式右端 $\to 0$, 即 $\left\|x^{(k+1)} - x_*\right\| \to 0$. 因此算法收敛. □

我们记向量 $e^{(k)} \triangleq x^{(k)} - x_*$ 为第 k 步迭代解 $x^{(k)}$ 的**误差向量**.

定理 6.4 (收敛性定理) 对任意迭代初始向量 $x^{(0)}$, 迭代算法 6.8 收敛的充要条件是 $\rho(G) < 1$.

证明. 必要性: 用反证法, 假设 $\rho(G) \ge 1$. 设 λ 为 G 的模最大的特征值, 即 $|\lambda| = \rho(G) \ge 1$. 令 $x \ne 0$ 为其对应的特征向量. 取迭代初始向量 $x^{(0)} = x_* + x$, 则

$$x^{(k)} - x_* = G(x^{(k-1)} - x_*) = \cdots = G^k(x^{(0)} - x_*) = G^k x = \lambda^k x,$$

不可能收敛到 0, 即算法不收敛. 故假设不成立, 因此 $\rho(G) < 1$.

充分性: 若 $\rho(G) < 1$, 则由引理 6.3 可知, 存在一个算子范数 $\|\cdot\|_\varepsilon$, 使得 $\|G\|_\varepsilon < 1$. 再由引理 6.5 可知, 算法收敛. □

定义 6.4 设 G 是迭代矩阵, 则迭代算法 6.8 的**平均收敛速度**定义为

$$R_k(G) \triangleq -\ln \|G^k\|^{\frac{1}{k}},$$

渐进收敛速度定义为

$$R(G) \triangleq \lim_{k\to\infty} R_k(G) = -\ln \rho(G).$$

☞ 平均收敛速度与迭代步数和所用的范数有关,但渐进收敛速度只依赖于迭代矩阵的谱半径.

定理 6.5 考虑迭代算法 6.8. 如果存在某个算子范数 $\|\cdot\|$ 使得 $\|G\| = q < 1$, 则

(1) $\|x^{(k)} - x_*\| \le q^k \|x^{(0)} - x_*\|$;

(2) $\|x^{(k)} - x_*\| \le \dfrac{q}{1-q}\|x^{(k)} - x^{(k-1)}\|$;

(3) $\|x^{(k)} - x_*\| \le \dfrac{q^k}{1-q}\|x^{(1)} - x^{(0)}\|$.

证明.

(1) 由 $x^{(k+1)} = Gx^{(k)} + g$ 和 $x_* = Gx_* + g$ 可得

$$x^{(k+1)} - x^{(k)} = G\left(x^{(k)} - x^{(k-1)}\right) \quad 和 \quad x^{(k+1)} - x_* = G\left(x^{(k)} - x_*\right).$$

因此有

$$\|x^{(k+1)} - x^{(k)}\| \le q\|x^{(k)} - x^{(k-1)}\|, \tag{6.9}$$

$$\|x^{(k+1)} - x_*\| \le q\|x^{(k)} - x_*\|. \tag{6.10}$$

反复利用结论 (6.10) 即可得 $\|x^{(k)} - x_*\| \le q^k \|x^{(0)} - x_*\|$.

(2) 由于

$$\begin{aligned}\|x^{(k+1)} - x^{(k)}\| &= \|x_* - x^{(k)} - (x_* - x^{(k+1)})\| \\ &\ge \|x_* - x^{(k)}\| - \|x_* - x^{(k+1)}\| \\ &\ge (1-q)\|x_* - x^{(k)}\|.\end{aligned}$$

结合 (6.22) 可得

$$\|x_* - x^{(k)}\| \le \frac{1}{1-q}\|x^{(k+1)} - x^{(k)}\| \le \frac{q}{1-q}\|x^{(k)} - x^{(k-1)}\|.$$

(3) 反复利用 (6.22) 即可得

$$\|x^{(k+1)} - x^{(k)}\| \le q^k \|x^{(1)} - x^{(0)}\|.$$

所以

$$\|x_* - x^{(k)}\| \le \frac{1}{1-q}\|x^{(k+1)} - x^{(k)}\| \le \frac{q^k}{1-q}\|x^{(1)} - x^{(0)}\|.$$

□

一般来说, 基于矩阵分裂的好的迭代算法应该满足: (1) $\rho(G)$ 很小; (2) $M^{-1}N$ 和 $M^{-1}b$ 容易计算, 或者以 M 为系数矩阵的线性方程组比较容易求解.

6.2.2 Jacobi 迭代

将矩阵 A 分裂为

$$A = D - L - U,$$

其中 D 为 A 的对角线部分, $-L$ 和 $-U$ 分别为 A 的严格下三角和严格上三角部分.

在矩阵分裂 $A = M - N$ 中取 $M = D, N = L + U$, 则可得 Jacobi 迭代算法:

$$x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b, \quad k = 0, 1, 2, \ldots. \tag{6.11}$$

迭代矩阵为

$$G_{\mathrm{J}} = D^{-1}(L+U).$$

写成分量形式即为

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{(k)}\right), \quad i = 1, 2, \ldots, n.$$

☞ 由于 Jacobi 迭代中 $x_i^{(k+1)}$ 的更新顺序与 i 无关, 即可以按顺序 $i = 1, 2, \ldots, n$ 计算, 也可以按顺序 $i = n, n-1, \ldots, 2, 1$ 计算, 或者乱序计算. 因此 Jacobi 迭代非常适合并行计算.

算法 6.1 求解线性方程组的 Jacobi 迭代算法

```
1: Choose an initial guess x^(0)
2: while not converge do
3:     for i = 1 to n do
4:         x_i^(k+1) = ( b_i − Σ_{j=1,j≠i}^{n} a_ij x_j^(k) ) / a_ii
5:     end for
6: end while
```

我们有时也将 Jacobi 迭代格式写为

$$x^{(k+1)} = x^{(k)} + D^{-1}(b - Ax^{(k)}) = x^{(k)} + D^{-1}r_k, \quad k = 0, 1, 2, \ldots,$$

其中 $r_k \triangleq b - Ax^{(k)}$ 是 k 次迭代后的残量.

下面给出求解二维离散 Poisson 方程 (6.5) 的 Jacobi 迭代算法.

算法 6.2 求解二维离散 Poisson 方程的 Jacobi 迭代算法

```
1: Choose an initial guess v^(0)
2: while not converge do
3:     for i = 1 to N do
4:         for j = 1 to N do
5:             u_{i,j}^(k+1) = ( h^2 f_{i,j} + u_{i+1,j}^(k) + u_{i−1,j}^(k) + u_{i,j+1}^(k) + u_{i,j−1}^(k) ) / 4
6:         end for
7:     end for
8: end while
```

6.2.3 Gauss-Seidel 迭代

在分裂 $A = M - N$ 中取 $M = D - L, N = U$, 即可得 Gauss-Seidel (G-S) 迭代算法:

$$x^{(k+1)} = (D - L)^{-1}Ux^{(k)} + (D - L)^{-1}b. \tag{6.12}$$

迭代矩阵为

$$G_{\text{GS}} = (D - L)^{-1}U.$$

将 G-S 迭代改写为

$$Dx^{(k+1)} = Lx^{(k+1)} + Ux^{(k)} + b,$$

即可得分量形式

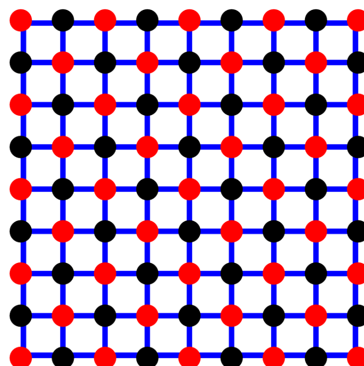
$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), \quad i = 1, 2, \ldots, n.$$

算法 6.3 求解线性方程组的 G-S 迭代算法

1: Choose an initial guess $x^{(0)}$
2: **while** not converge **do**
3:     **for** $i = 1$ to n **do**
4:         $x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)$
5:     **end for**
6: **end while**

由于 G-S 算法能充分利用已经获得的最新数据, 故通常比 Jacobi 算法快速. 与 Jacobi 迭代算法类似, 我们也可以给出求解问题 (6.5) 的 G-S 迭代算法. 注意, 此时未知量的更新是按自然顺序进行的. 由于在对未知量进行更新时必须按照 $i = 1, 2, \ldots, n$ 的顺序进行, 因此不适合并行计算.

下面我们介绍一种适合并行计算的更新顺序: 红黑排序, 即将二维网格点依次做红黑记号, 如右图所示.



在计算过程中, 对未知量的值进行更新时, 我们可以先更新红色节点, 此时所使用的只是黑色节点的数据, 然后再更新黑色节点, 这时使用的是红色节点的数据. 于是我们得到下面红黑排序 G-S 迭代算法.

由于在更新红点时, 各个点之间是相互独立的, 因此可以并行计算. 同样, 在更新黑点时, 各个点之间也是相互独立的, 因此也可以并行计算.

算法 6.4 求解二维离散 Poisson 方程的红黑排序 G-S 迭代算法

1: Choose an initial guess $v^{(0)}$
2: **while** not converge **do**
3:     **for** (i, j) 为红色节点 **do**
4:         $u_{i,j}^{(k+1)} = \frac{1}{4}\left(h^2 f_{i,j} + u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k)}\right)$

5: **end for**
6: **for** (i,j) 为黑色节点 **do**
7: $u_{i,j}^{(k+1)} = \frac{1}{4}\left(h^2 f_{i,j} + u_{i+1,j}^{(k+1)} + u_{i-1,j}^{(k+1)} + u_{i,j+1}^{(k+1)} + u_{i,j-1}^{(k+1)}\right)$
8: **end for**
9: **end while**

6.2.4 SOR 迭代

在 G-S 算法的基础上, 我们可以通过引入一个松弛参数 ω 来加快收敛速度. 这就是 SOR (Successive Overrelaxation) 算法 [45]. 该方法的基本思想是将 G-S 算法中的第 $k+1$ 步近似解与第 k 步近似解做一个加权平均, 从而给出一个新的近似解, 即

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega\left(D^{-1}(Lx^{(k+1)} + Ux^{(k)}) + D^{-1}b\right). \tag{6.13}$$

整理后即为

$$x^{(k+1)} = (D-\omega L)^{-1}\left((1-\omega)D + \omega U\right)x^{(k)} + \omega(D-\omega L)^{-1}b, \tag{6.14}$$

其中 ω 称为**松弛参数** (relaxation parameter) . 当 $\omega = 1$ 时, SOR 即为 G-S 算法, 当 $\omega < 1$ 时, 称为**低松弛** (under relaxation) 算法, 当 $\omega > 1$ 时, 称为**超松弛** (over relaxation) 算法. SOR 算法曾经在很长一段时间内是科学计算中求解线性方程组的首选方法. 在大多数情况下, 当 $\omega > 1$ 时会取得比较好的收敛效果.

SOR 的迭代矩阵为

$$G_{\text{SOR}} = (D-\omega L)^{-1}\left((1-\omega)D + \omega U\right),$$

对应的矩阵分裂为

$$M = \frac{1}{\omega}D - L, \quad N = \frac{1-\omega}{\omega}D + U.$$

由 (6.14) 可得 SOR 迭代的分量形式为

$$\begin{aligned} x_i^{(k+1)} &= (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right) \\ &= x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right) \end{aligned}$$

算法 6.5 求解线性方程组的 SOR 迭代算法

1: Choose an initial guess $x^{(0)}$ and parameter ω
2: **while** not converge **do**
3: **for** $i = 1$ to n **do**
4: $x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)$
5: **end for**
6: **end while**

SOR 算法最大的优点是引入了松弛参数 ω, 通过选取适当的 ω 可以大大提高算法的收敛速度. 但是 SOR 算法最大的难点就是如何选取最优的参数.

算法 6.6 求解二维离散 Poisson 方程的红黑排序 SOR 迭代算法

1: Choose an initial guess $v^{(0)}$ and parameter ω
2: **while** not converge **do**
3:     **for** (i,j) 为红色节点 **do**
4:         $u_{i,j}^{(k+1)} = (1-\omega)v_{i,j}^{(k)} + \omega(h^2 f_{i,j} + u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k)})/4$
5:     **end for**
6:     **for** (i,j) 为黑色节点 **do**
7:         $u_{i,j}^{(k+1)} = (1-\omega)v_{i,j}^{(k)} + \omega(h^2 f_{i,j} + u_{i+1,j}^{(k+1)} + u_{i-1,j}^{(k+1)} + u_{i,j+1}^{(k+1)} + u_{i,j-1}^{(k+1)})/4$
8:     **end for**
9: **end while**

6.2.5 SSOR 迭代算法

将 SOR 算法中的 L 和 U 相交换, 即可得迭代格式

$$x^{(k+1)} = (D-\omega U)^{-1}\left((1-\omega)D + \omega L\right)x^{(k)} + \omega(D-\omega U)^{-1}b,$$

将这个迭代格式与 SOR 相结合, 就可以得到下面的两步迭代算法

$$\begin{cases} x^{(k+\frac{1}{2})} = (D-\omega L)^{-1}\left[(1-\omega)D+\omega U\right]x^{(k)} + \omega(D-\omega L)^{-1}b \\ x^{(k+1)} = (D-\omega U)^{-1}\left[(1-\omega)D+\omega L\right]x^{(k+\frac{1}{2})} + \omega(D-\omega U)^{-1}b \end{cases}$$

这就是 SSOR 迭代 (对称超松弛) 算法, 相当于将 L 与 U 同等看待, 交替做两次 SOR 迭代.

消去中间迭代向量 $x^{(k+\frac{1}{2})}$, 可得

$$x^{(k+1)} = G_{\text{SSOR}}x^{(k)} + g,$$

其中迭代矩阵

$$G_{\text{SSOR}} = (D-\omega U)^{-1}\left[(1-\omega)D+\omega L\right](D-\omega L)^{-1}\left[(1-\omega)D+\omega U\right].$$

对应的矩阵分裂为

$$\begin{aligned} M &= \frac{1}{\omega(2-\omega)}\left[D - \omega(L+U) + \omega^2 LD^{-1}U\right] \\ &= \frac{1}{\omega(2-\omega)}(D-\omega L)D^{-1}(D-\omega U), \\ N &= \frac{1}{\omega(2-\omega)}\left[(1-\omega)D+\omega L\right]D^{-1}\left[(1-\omega)D+\omega U\right]. \end{aligned}$$

☞ 对于某些特殊问题, SOR 算法不收敛, 但仍然可能构造出收敛的 SSOR 算法.

☞ 一般来说, SOR 算法的渐进收敛速度对参数 ω 比较敏感, 但 SSOR 对参数 ω 不太敏感.

算法 6.7 SSOR 算法

1: Choose an initial guess $v^{(0)}$ and parameter ω
2: **while** not converge **do**
3:     **for** $i = 1$ to n **do**
4:         $x_i^{(k+\frac{1}{2})} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+\frac{1}{2})} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)$
5:     **end for**
6:     **for** $i = n$ to 1 **do**
7:         $x_i^{(k+1)} = (1-\omega)x_i^{(k+\frac{1}{2})} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+\frac{1}{2})} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k+1)}\right)$
8:     **end for**
9: **end while**

6.2.6 AOR 迭代

Hadjidimos 于 1978 年提出了 AOR (Accelerated over-relaxation, 快速松弛) 算法, 迭代矩阵为

$$G_{\mathrm{AOR}} = (D-\gamma L)^{-1}\big[(1-\omega)D + (\omega-\gamma)L + \omega U\big],$$

其中 γ 和 ω 为松弛参数. 对应的矩阵分解为

$$M = \frac{1}{\omega}(D-\gamma L), \quad N = \frac{1}{\omega}[(1-\omega)D + (\omega-\gamma)L + \omega U].$$

(1) 当 $\gamma = \omega$ 时, AOR 算法即为 SOR 算法;
(2) 当 $\gamma = \omega = 1$ 时, AOR 算法即为 G-S 算法;
(3) 当 $\gamma = 0, \omega = 1$ 时, AOR 算法即为 Jacobi 算法.

☞ 与 SSOR 类似, 我们也可以定义 SAOR 算法.

6.2.7 Richardson 算法

Richardson 算法是一类形式非常简单的算法, 其迭代格式为

$$x^{(k+1)} = x^{(k)} + \omega(b - Ax^{(k)}), \quad k = 0, 1, 2, \ldots.$$

其对应的矩阵分解为

$$M = \frac{1}{\omega}I, \quad N = \frac{1}{\omega}I - A,$$

迭代矩阵为

$$G_{\mathrm{R}} = I - \omega A.$$

定理 6.6 设 $A \in \mathbb{R}^{n\times n}$ 是对称正定矩阵, λ_1 和 λ_n 分别是 A 的最大和最小特征值, 则 *Richardson* 算法收敛当且仅当

$$0 < \omega < \frac{1}{\lambda_1}.$$

最优参数为

$$\omega_* = \arg\min_{\omega} \rho(G_{\mathrm{R}}) = \frac{2}{\lambda_1 + \lambda_n},$$

即当 $\omega = \omega_*$ 时, 迭代矩阵的谱半径达到最小, 且有

$$\rho(G_{\mathrm{R}}) = \begin{cases} 1 - \omega\lambda_n & \text{if } \omega \le \omega_* \\ \dfrac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} = \dfrac{\kappa(A) - 1}{\kappa(A) + 1} & \text{if } \omega = \omega_* \\ \omega\lambda_1 - 1 & \text{if } \omega \ge \omega_*. \end{cases}$$

如果在每次迭代时取不同的参数, 即

$$x^{(k+1)} = x^{(k)} + \omega_k(b - Ax^{(k)}), \quad k = 0, 1, 2, \dots,$$

则称为 nonstationary Richardson 算法.

6.2.8 分块迭代算法

前面介绍的迭代算法可以推广到分块情形. 将 A 写成如下的分块形式:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{bmatrix}.$$

A_{11} A_{22} A_{pp}

设 $A = D - L - U$, 其中 $D, -L, -U$ 分别是 A 的块对角, 块严格下三角和块严格上三角矩阵. 则相应的分块 Jacobi, 分块 Gauss-Seidel 和分块 SOR 算法分别为

- 分块 Jacobi 迭代:

$$A_{ii}\boldsymbol{x}_i^{(k+1)} = \boldsymbol{b}_i - \sum_{j=1, j\ne i}^{p} A_{ij}\boldsymbol{x}_j^{(k)}, \quad i = 1, 2, \dots, p.$$

- 分块 Gauss-seidel 迭代:

$$A_{ii}\boldsymbol{x}_i^{(k+1)} = \boldsymbol{b}_i - \sum_{j=1}^{i-1} A_{ij}\boldsymbol{x}_j^{(k+1)} - \sum_{j=i+1}^{p} A_{ij}\boldsymbol{x}_j^{(k)}, \quad i = 1, 2, \dots, p.$$

- 分块 SOR 迭代:

$$\boldsymbol{x}_i^{(k+1)} = (1-\omega)\boldsymbol{x}_i^{(k)} + \omega A_{ii}^{-1}\left(\boldsymbol{b}_i - \sum_{j=1}^{i-1} A_{ij}\boldsymbol{x}_j^{(k+1)} - \sum_{j=i+1}^{p} A_{ij}\boldsymbol{x}_j^{(k)}\right),$$

$$i = 1, 2, \dots, p.$$

6.3 迭代算法的收敛性

对于矩阵分裂迭代算法, 其收敛的充要条件是迭代矩阵的谱半径小于 1. 当谱半径不可求时, 如果迭代矩阵的某个算子范数小于 1, 则算法也收敛.

6.3.1 二维离散 Poisson 方程的 Jacobi, G-S 和 SOR 算法的收敛性

考虑这些算法的收敛性, 只需研究相应的迭代矩阵的谱半径即可. 对于二维离散 Poisson 方程, 系数矩阵为

$$A = T = I \otimes T_n + T_n \otimes I.$$

故 Jacobi 算法的迭代矩阵为

$$G_{\rm J} = D^{-1}(L+U) = (4I)^{-1}(4I-T) = I - T/4. \tag{6.15}$$

由于 T 的特征值为

$$\lambda_i + \lambda_j = 2\left(1 - \cos\frac{\pi i}{n+1}\right) + 2\left(1 - \cos\frac{\pi j}{n+1}\right) = 4 - 2\left(\cos\frac{\pi i}{n+1} + \cos\frac{\pi j}{n+1}\right),$$

所以 $G_{\rm J}$ 的特征值为

$$1 - (\lambda_i + \lambda_j)/4 = \frac{1}{2}\left(\cos\frac{\pi i}{n+1} + \cos\frac{\pi j}{n+1}\right).$$

故

$$\rho(G_{\rm J}) = \frac{1}{2}\max_{i,j}\left\{\left|\cos\frac{\pi i}{n+1} + \cos\frac{\pi j}{n+1}\right|\right\} = \cos\frac{\pi}{n+1} < 1,$$

即 Jacobi 算法是收敛的.

注意当 n 越来越大时, $\kappa(T) \to \infty$, 即 T 越来越病态, 此时 $\rho(G_{\rm J}) \to 1$, 即 Jacobi 算法收敛越来越慢.

☞ 通常, 问题越病态, 迭代算法的收敛就越慢.

关于 G-S 算法和 SOR 算法, 我们有下面的结论.

性质 6.1 *设 $G_{\rm GS}$ 和 $G_{\rm SOR}$ 分别表示求解二维 Poisson 方程的红黑排序的 G-S 算法和 SOR 算法的迭代矩阵, 则有*

$$\rho(G_{\rm GS}) = \rho(G_{\rm J})^2 = \cos^2\frac{\pi}{n+1} < 1 \tag{6.16}$$

$$\rho(G_{\rm SOR}) = \frac{\cos^2\dfrac{\pi}{n+1}}{\left(1+\sin\dfrac{\pi}{n+1}\right)^2} < 1, \quad \omega = \frac{2}{1+\sin\dfrac{\pi}{n+1}}. \tag{6.17}$$

在上述结论中, SOR 算法中的 ω 是最优参数, 即此时的 $\rho(G_{\rm SOR})$ 最小. 由 Taylor 公式可知, 当 n 很大时, 有

$$\rho(G_{\rm J}) = \cos\frac{\pi}{n+1} \approx 1 - \frac{\pi^2}{2(n+1)^2} = 1 - O\left(\frac{1}{n^2}\right),$$

$$\rho(G_{\text{SOR}}) = \frac{\cos^2 \dfrac{\pi}{n+1}}{\left(1+\sin\dfrac{\pi}{n+1}\right)^2} \approx 1 - \frac{2\pi}{n+1} = 1 - O\left(\frac{1}{n}\right).$$

由于当 n 很大时有

$$\left(1-\frac{1}{n}\right)^k \approx 1 - \frac{k}{n} = 1 - \frac{kn}{n^2} \approx \left(1-\frac{1}{n^2}\right)^{kn},$$

即 SOR 算法迭代 k 步后误差的减小量与 Jacobi 算法迭代 kn 步后误差减小量差不多. 因此, 对于二维离散 Poisson 方程, 当 SOR 算法取最优参数时, 收敛速度大约是 Jacobi 算法的 n 倍.

☞ 这里需要指出的是, 对于一般线性方程组, 上述结论不一定成立.

由于 $\rho(G_{\text{GS}}) = \rho(G_{\text{J}})^2$, 因此, 对于二维离散 Poisson 方程, G-S 算法的收敛速度大约是 Jacobi 算法的 2 倍.

☞ 事实上, 当 n 很大时, 这三个算法的收敛速度都很慢.

例 6.1 已知二维 Poisson 方程

$$\begin{cases} -\Delta u(x,y) = -1, & (x,y) \in \Omega \\ u(x,y) = \dfrac{x^2+y^2}{4}, & (x,y) \in \partial\Omega \end{cases}$$

其中 $\Omega = (0,1) \times (0,1)$. 该方程的解析解是 $u(x,y) = \dfrac{x^2+y^2}{4}$. 用五点差分格式离散后得到一个线性方程组, 分别用 Jacobi, G-S 和 SOR 算法计算这个方程组的解, 并比较收敛效果.

解. 见程序 `Jacobi_GS_SOR_Poisson.m`. 下图中画出了 $N = 16, 32, 64, 128$ 时, 这三种算法的相对误差下降情况.

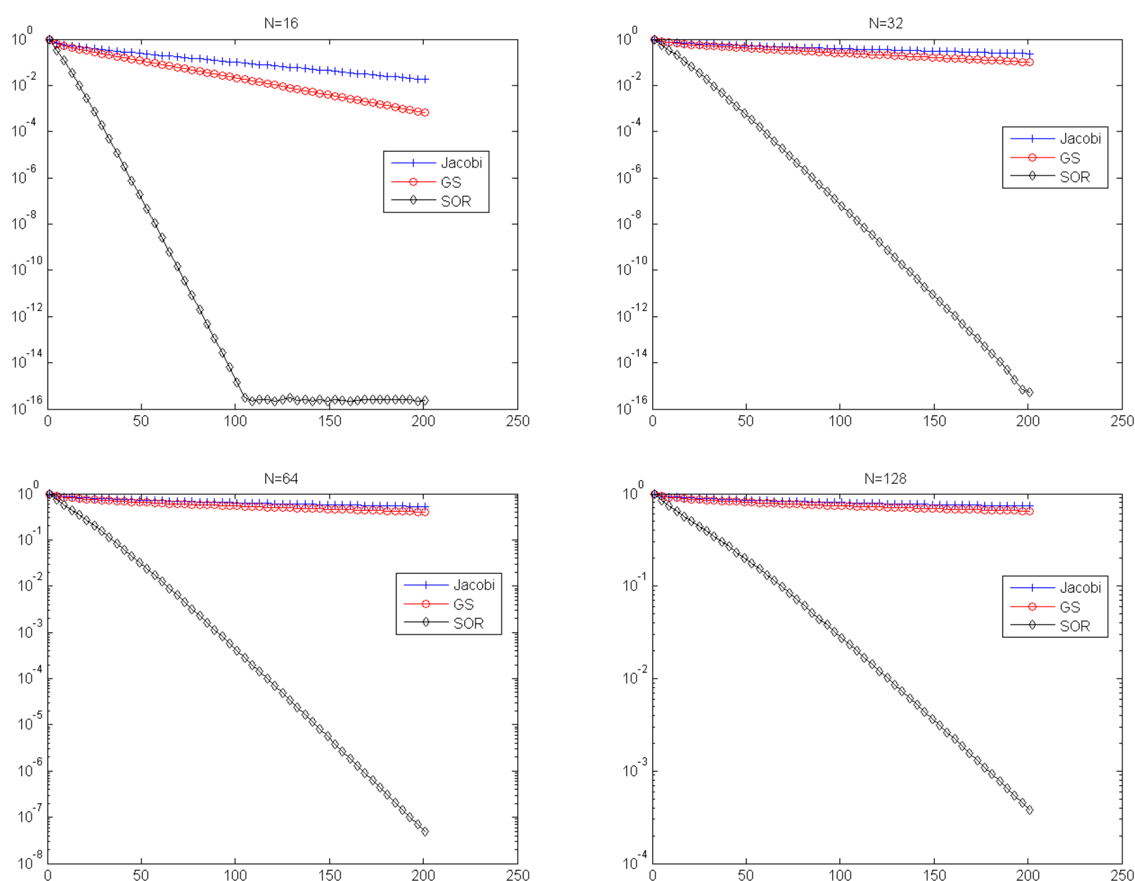


图 6.1 Jacobi, G-S 和 SOR 算法的相对误差下降情况.

□

6.3.2 不可约对角占优

我们首先给出不可约矩阵的定义.

定义 6.5 设 $A \in \mathbb{R}^{n\times n}$, 如果存在置换矩阵 P, 使得 PAP^T 为块上三角矩阵, 即

$$PAP^T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix},$$

其中 $A_{11} \in \mathbb{R}^{k\times k}$ $(1 \le k < n)$ 则称 A 为**可约矩阵**, 否则称为**不可约矩阵**.

下面的定理可用于判别一个矩阵是否可约.

定理 6.7 设 $A \in \mathbb{R}^{n\times n}$, 指标集 $\mathbb{Z}_n = \{1, 2, \ldots, ..., n\}$. 则 A 可约的充要条件是存在非空指标集 $J \subset \mathbb{Z}_n$ 且 $J \neq \mathbb{Z}_n$, 使得

$$a_{ij} = 0, \quad i \in J \text{ 且 } j \in \mathbb{Z}_n \setminus J.$$

这里 $\mathbb{Z}_n \setminus J$ 表示 J 在 $\mathbb{Z}_n$ 中的补集.

证明. 留作练习. □

我们知道,严格对角占优矩阵是非奇异的,见定理 1.18. 如果 A 是不可约的弱对角占优矩阵,则可以同样证明 A 是非奇异的.

定理 6.8 设 $A \in \mathbb{R}^{n\times n}$ 是不可约的弱对角占优矩阵, 则 A 非奇异.

证明. 我们采用反证法. 假设 A 是奇异的, 即存在特征值 $\lambda = 0$. 设其对应的非零特征值向量为 $x = [x_1, x_2, \ldots, x_n]^T$, 满足 $\|x\|_\infty = 1$, 即 $|x_j| \le 1, j = 1, 2, \ldots, n$, 且存在指标 r 使得 $|x_r| = 1$. 定义指标集

$$J = \{r \in \mathbb{Z}_n : |x_r| = 1\}.$$

由 $Ax = \lambda x = 0$ 可知, 任意的 $r \in J$, 有

$$a_{rr}x_r = -\sum_{j=1, j\neq r}^{n} a_{rj}x_j.$$

所以

$$|a_{rr}| \le \frac{1}{|x_r|}\sum_{j=1, j\neq r}^{n} |a_{rj}|\cdot|x_j| \le \sum_{j=1, j\neq r}^{n} |a_{rj}|, \tag{6.18}$$

其中等号成立当且仅当

$$a_{rj} = 0, \quad j \in \mathbb{Z}_n \setminus J. \tag{6.19}$$

由于 A 是弱对角占优的, 故

$$|a_{rr}| = \sum_{j=1, j\neq r}^{n} |a_{rj}|.$$

所以结论 (6.19) 对所有的 $r \in J$ 都成立.

显然, J 是非空的. 下面我们说明 $J \neq \mathbb{Z}_n$. 否则的话, 如果 $J = \mathbb{Z}_n$, 则由 (6.18) 可知 A 不可能是弱对角占优的. 所以 J 非空且 $J \neq \mathbb{Z}_n$. 由定理 6.7 可知 A 是可约的, 这与条件 A 不可约矛盾. 所以, $\lambda = 0$ 不是 A 的特征值, 即 A 非奇异. □

定理 6.9 设 $A \in \mathbb{R}^{n\times n}$, 若 A 严格对角占优, 则 Jacobi 算法和 G-S 算法都收敛, 且

$$\|G_{\rm GS}\|_\infty \le \|G_{\rm J}\|_\infty < 1.$$

证明. 首先证明 $\|G_{\rm J}\|_\infty < 1$. 由于 A 严格行对角占优, 故 $\sum_{j\neq i} |a_{ij}|/|a_{ii}| < 1$. 所以

$$\|G_{\rm J}\|_\infty = \|D^{-1}(L+U)\|_\infty = \max_{1\le i\le n}\sum_{j\neq i}\frac{|a_{ij}|}{|a_{ii}|} < 1.$$

下面证明 $\|G_{\rm GS}\|_\infty \le \|G_{\rm J}\|_\infty$, 只需证明 $|G_{\rm GS}|e \le |G_{\rm J}|e$ 即可, 其中 $e = [1, 1, \ldots, 1]^T$.

令 $\tilde{L} = D^{-1}L$ 和 $\tilde{U} = D^{-1}U$. 则 $\tilde{L}$ 是绝对下三角矩阵, 故 $\tilde{L}^n = 0$, 所以

$$(I - \tilde{L})^{-1} = I + \tilde{L} + \tilde{L}^2 + \cdots + \tilde{L}^{n-1}.$$

于是

$$|G_{\rm GS}|e = |(D-L)^{-1}U|e = |(I-\tilde{L})^{-1}\tilde{U}|e$$

$$
\begin{aligned}
&= |(I + \tilde{L} + \tilde{L}^2 + \cdots + \tilde{L}^{n-1})\tilde{U}|e \\
&\le (I + |\tilde{L}| + |\tilde{L}|^2 + \cdots + |\tilde{L}|^{n-1})|\tilde{U}|e \\
&= (I - |\tilde{L}|)^{-1}|\tilde{U}|e. \qquad (6.20)
\end{aligned}
$$

由 A 的严格行对角占优性可知 $1 - \sum_{j\neq i} \dfrac{|a_{ij}|}{|a_{ii}|} > 0$, 即 $(I - |\tilde{L}| - |\tilde{U}|)e > 0$. 两边同乘 $|\tilde{L}| \ge 0$ 可得

$$
\begin{aligned}
0 \le |\tilde{L}|(I - |\tilde{L}| - |\tilde{U}|)e &= (|\tilde{L}| - |\tilde{L}|^2 + |\tilde{U}| - |\tilde{L}| \cdot |\tilde{U}| - |\tilde{U}|)e \\
&= \big((I - |\tilde{L}|)(|\tilde{L}| + |\tilde{U}|) - |\tilde{U}|\big)e,
\end{aligned}
$$

即 $|\tilde{U}|e \le (I - |\tilde{L}|)(|\tilde{L}| + |\tilde{U}|)e$. 两边同乘 $(I - |\tilde{L}|)^{-1} \ge 0$ 可得

$$
(I - |\tilde{L}|)^{-1}|\tilde{U}|e \le (|\tilde{L}| + |\tilde{U}|)e.
$$

又 $|\tilde{L}| + |\tilde{U}| = |D^{-1}(L + U)| = |G_{\rm J}|$, 由 (6.20) 可知 $|G_{\rm GS}|e \le |G_{\rm J}|e$, 即定理结论成立. □

☞ 当 A 是严格列对角占优时, 该结论也成立.

定理 6.10 设 $A \in \mathbb{R}^{n\times n}$, 若 A 是弱对角占优且不可约, 则 *Jacobi* 算法和 *G-S* 算法都收敛, 且 $\rho(G_{\rm GS}) < \rho(G_{\rm J}) < 1$.

证明. 略. (可参见 [44]) □

☞ 二维离散 Poisson 方程是弱行对角占优且不可约, 故对 Jacobi 算法和 G-S 算法都收敛.

☞ 上述定理中的结论对一般矩阵并不成立: 对某些矩阵, Jacobi 算法收敛, 但 G-S 算法却不一定收敛.

6.3.3 对称正定矩阵

引理 6.6 设 $A \in \mathbb{C}^{n\times n}$ *Hermite* 对称, 且 $A = M - N$ 是 A 的一个矩阵分裂, 则 $M^* + N$ 也是 *Hermite* 对称, 且对任意 $x \in \mathbb{C}^n$ 有

$$
x^*Ax - \tilde{x}^*A\tilde{x} = u^*(M^* + N)u,
$$

其中 $\tilde{x} = M^{-1}Nx$, $u = x - \tilde{x}$.

证明. 由于 A Hermite 对称, 所以 $M^* + N = M^* + M - A$ 也 Hermite 对称.

由于 $\tilde{x} = M^{-1}Nx$, 所以 $M\tilde{x} = Nx$, 因此

$$
\begin{aligned}
Mu &= Mx - M\tilde{x} = Mx - Nx = Ax, \\
Nu &= Nx - N\tilde{x} = M\tilde{x} - N\tilde{x} = A\tilde{x}.
\end{aligned}
$$

由 $M^* + N$ 的对称性可知 $M = M^* + N - N^*$. 又 $(Nx)^* = (M\tilde{x})^*$, 所以

$$
x^*Ax - \tilde{x}^*A\tilde{x} = x^*Mu - \tilde{x}^*Nu
$$

$$
\begin{aligned}
&= x^*(M^* + N - N^*)u - \tilde{x}^* N u \\
&= x^* M^* u - x^* N^* u + x^* N u - \tilde{x}^* N u \\
&= x^* M^* u - \tilde{x}^* M^* u + u^* N u \\
&= u^*(M^* + N)u.
\end{aligned}
$$

□

引理 6.7 设 $A \in \mathbb{R}^{n\times n}$ 对称, 且 $A = M - N$ 是 A 的一个矩阵分裂.

(1) 如果 A 和 $M^T + N$ 都是正定矩阵, 则 M 非奇异且 $\rho(M^{-1}N) < 1$;

(2) 如果 $\rho(M^{-1}N) < 1$ 且 $M^T + N$ 正定, 则 A 正定.

证明. (1) 首先证明 M 是非奇异的. 用反证法, 假设 M 奇异, 则存在非零向量 x 使得 $Mx = 0$, 故有 $x^T M^T = (Mx)^T = 0$. 又 A 对称正定, 所以 $x^T(M + M^T - A)x = -x^T Ax < 0$, 与 $M + M^T - A$ 对称正定矛盾, 故 M 非奇异.

下面证明 $\rho(M^{-1}N) < 1$. 设 $\lambda \in \mathbb{C}$ 是 $M^{-1}N$ 的一个特征值, 对应的特征向量为 $x \neq 0$, 即

$$
M^{-1}Nx = \lambda x.
$$

我们首先说明 $\lambda \neq 1$. 假设 $\lambda = 1$, 则可得 $Mx = Nx$, 即 $Ax = 0$. 由于 A 非奇异, 所以 $x = 0$, 矛盾.

令 $\tilde{x} = M^{-1}Nx = \lambda x, u = x - \tilde{x} = (1 - \lambda)x$. 则由引理 6.6 可得

$$
(1 - |\lambda|^2)x^* Ax = |1 - \lambda|^2 x^*(M^T + N)x.
$$

由于 A 和 $M^T + N$ 都是正定矩阵, 所以上式右端为正, 故 $|\lambda| < 1$. 因此 $\rho(M^{-1}N) < 1$.

(2) 反证法. 假设 A 不是正定的. 由于 $\rho(M^{-1}N) < 1$, 所以 $A = M(I - M^{-1}N)$ 非奇异, 因此存在 $x^{(0)} \in \mathbb{R}^n$, 使得

$$
\eta \triangleq \left(x^{(0)}\right)^T Ax^{(0)} < 0.
$$

以 $x^{(0)}$ 为初始点, 构造迭代序列

$$
x^{(k)} = M^{-1}Nx^{(k-1)}, \quad k = 1, 2, \ldots.
$$

由 $\rho(M^{-1}N) < 1$ 可知

$$
\lim_{k\to\infty} x^{(k)} = \lim_{k\to\infty} (M^{-1}N)^k x^{(0)} = 0. \tag{6.21}
$$

令 $u^{(k)} = x^{(k-1)} - x^{(k)}$, 则由引理 6.6 可得

$$
\left(x^{(k-1)}\right)^T Ax^{(k-1)} - \left(x^{(k)}\right)^T Ax^{(k)} = \left(u^{(k)}\right)^T (M^T + N)u^{(k)}.
$$

由于 $M^T + N$ 对称正定, 上式右端非负, 所以

$$
\left(x^{(k)}\right)^T Ax^{(k)} \leq \left(x^{(k-1)}\right)^T Ax^{(k-1)}.
$$

依此类推, 可得

$$
\left(x^{(k)}\right)^T Ax^{(k)} \leq \left(x^{(0)}\right)^T Ax^{(0)} = \eta < 0.
$$

这与 (6.21) 矛盾. 因此 A 一定是正定的. □

我们首先给出 SOR 迭代收敛的一个必要条件.

定理 6.11 *对于 SOR 算法, 有 $\rho(G_{\rm SOR}) \geq |1-\omega|$, 故 SOR 算法收敛的必要条件是 $0<\omega<2$.*

证明. SOR 算法的迭代矩阵为

$$G_{\rm SOR} = (D-\omega L)^{-1}\left((1-\omega)D+\omega U\right) = (I-\omega\tilde{L})^{-1}((1-\omega)I+\omega\tilde{U}).$$

所以 $G_{\rm SOR}$ 的行列式为

$$\begin{aligned}\det\left(G_{\rm SOR}\right) &= \det\left((I-\omega\tilde{L})^{-1}\right)\cdot\det((1-\omega)I+\omega\tilde{U}) \\ &= \left(\det(I-\omega\tilde{L})\right)^{-1}\cdot(1-\omega)^n \\ &= (1-\omega)^n.\end{aligned}$$

设 $G_{\rm SOR}$ 的特征为 $\lambda_1,\lambda_2,\ldots,\lambda_n$, 则

$$\lambda_1\lambda_2\cdots\lambda_n = \det\left(G_{\rm SOR}\right) = (1-\omega)^n,$$

故至少有一个特征值的绝对值不小于 $|1-\omega|$, 即 $\rho(G_{\rm SOR}) \geq |1-\omega|$.

若 SOR 收敛, 则 $\rho(G_{\rm SOR})<1$, 因此 $|1-\omega|<1$, 即 $0<\omega<2$. □

定理 6.12 设 $A\in\mathbb{R}^{n\times n}$ 对称正定.

(1) 若 $2D-A$ 正定, 则 *Jacobi* 迭代收敛.

(2) 若 $0<\omega<2$, 则 *SOR* 和 *SSOR* 收敛.

(3) *G-S* 迭代收敛.

证明. 留作练习 (利用引理 6.7). □

☞ 若系数矩阵对称正定, 则 SOR 算法收敛的充要条件是 $0<\omega<2$.

☞ 对于二维离散 Poisson 方程, 其系数矩阵是对称正定的, 故当 $0<\omega<2$ 时, SOR 算法收敛.

定理 6.13 设 $A\in\mathbb{R}^{n\times n}$ 对称.

(1) 若 $2D-A$ 正定且 *Jacobi* 迭代收敛, 则 A 正定;

(2) 若 D 正定, 且存在 $\omega\in(0,2)$ 使得 *SOR* (或 *SSOR*) 收敛, 则 A 正定;

(3) 若 D 正定, 且 *G-S* 迭代收敛, 则 A 正定.

证明. 留作练习 (利用引理 6.7). □

6.3.4 相容次序矩阵

针对一类特殊的矩阵, 这三种迭代算法的谱半径之间存在一种特殊的关系.

定义 6.6 设 $A \in \mathbb{R}^{n\times n}$, 如果存在一个置换矩阵 P, 使得

$$PAP^T = \begin{bmatrix} D_1 & F \\ E & D_2 \end{bmatrix}, \tag{6.22}$$

其中 D_1, D_2 为对角矩阵, 则称 A 具有性质 A .

例 6.2 对于二维离散 Poisson 方程, 系数矩阵 T_{N^2} 具有性质 A. 事实上, 设 $\tilde{T}_{N^2}$ 为模型问题采用红黑排序后的系数矩阵, 则 $\tilde{T}_{N^2}$ 具有 (6.22) 的结构.

我们首先给出一个性质.

引理 6.8 设 $B \in \mathbb{R}^{n\times n}$ 具有下面的结构

$$B = \begin{bmatrix} 0 & B_{12} \\ B_{21} & 0 \end{bmatrix},$$

令 B_L 和 B_U 分别表示 B 的下三角和上三角部分, 则

(1) 若 μ 是 B 的特征值, 则 $-\mu$ 也是 B 的特征值;

(2) $B(\alpha)$ 的特征值与 α 无关, 其中

$$B(\alpha) = \alpha B_L + \frac{1}{\alpha} B_U, \quad \alpha \neq 0.$$

证明. (1) 若 $[x, y]$ 是 B 的对应于 μ 的特征向量, 则 $[x, -y]$ 是 B 对应于 $-\mu$ 的特征向量.

(2) 由于 $\alpha \neq 0$, 我们有

$$\begin{bmatrix} I & 0 \\ 0 & \alpha I \end{bmatrix}^{-1} B(\alpha) \begin{bmatrix} I & 0 \\ 0 & \alpha I \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & \frac{1}{\alpha} I \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{\alpha} B_{12} \\ \alpha B_{21} & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \alpha I \end{bmatrix} = \begin{bmatrix} 0 & B_{12} \\ B_{21} & 0 \end{bmatrix},$$

故引理结论成立. □

☞ 由引理 6.8 中结论 (2) 可知, $B(\alpha) + \beta I$ 的特征值也与 α 无关, 其中 β 为任意常数.

☞ 该结论可以推广到块三对角形式, 见习题 6.6.

设 $A \in \mathbb{R}^{n\times n}$ 的对角线元素全不为零, 记 $\tilde{L} = D^{-1}L, \tilde{U} = D^{-1}U$.

定义 6.7 设 $A \in \mathbb{R}^{n\times n}$ 的对角线元素全不为零, $A = D(I - \tilde{L} - \tilde{U})$. 若矩阵 $G(\alpha) = \alpha\tilde{L} + \frac{1}{\alpha}\tilde{U}$ 的特征值与 α 无关, 则称 A 具有相容次序.

☞ 设 A 的对角线元素全不为零, 若 A 具有性质 A, 则存在置换矩阵 P, 使得 PAP^T 具有相容次序.

例 6.3 设 D_i 是非奇异的对角矩阵, 则任意块三对角矩阵

$$\begin{bmatrix} D_1 & A_1 & & \\ B_1 & \ddots & \ddots & \\ & \ddots & \ddots & A_{N-1} \\ & & B_{N-1} & D_N \end{bmatrix}$$

都有相容次序 (参见习题 6.6).

定理 6.14 设 A 有相容次序且 $\omega \neq 0$, 则下列命题成立

(1) *Jacobi* 迭代矩阵 G_{J} 的特征值正负成对出现;

(2) 若 μ 是 G_{J} 的特征值且 λ 满足

$$(\lambda + \omega - 1)^2 = \lambda \omega^2 \mu^2, \tag{6.23}$$

则 λ 是 *SOR* 迭代矩阵 G_{SOR} 的一个特征值;

(3) 反之, 若 $\lambda \neq 0$ 是 G_{SOR} 的一个特征值且 μ 满足 (6.23), 则 μ 是 G_{J} 的一个特征值.

证明. (1) 易知 $G_{\text{J}} = G(1)$. 设 μ 是 $G(1)$ 的特征值, 因为 $G(-1) = -G(1)$, 所以 $-\mu$ 是 $G(-1)$ 的特征值. 又 $G(\alpha)$ 的特征值与 α 无关, 故 $-\mu$ 也是 $G(1)$ 的特征值, 所以命题结论成立.

(2) 若 $\lambda = 0$, 则由 (6.23) 可知 $\omega = 1$, 此时 $G_{\text{SOR}} = (I - \tilde{L})^{-1}\tilde{U}$ 为奇异矩阵, 故 $\lambda = 0$ 是其特征值, 即命题结论成立.

若 $\lambda \neq 0$, 则 G_{SOR} 的特征多项式为

$$\begin{aligned}
\det(\lambda I - G_{\text{SOR}}) &= \det\left(\lambda I - (I - \omega\tilde{L})^{-1}\left((1-\omega)I + \omega\tilde{U}\right)\right) \\
&= \det\left((I - \omega\tilde{L})^{-1}\right) \cdot \det\left(\lambda(I - \omega\tilde{L}) - (1-\omega)I - \omega\tilde{U}\right) \\
&= \det\left((\lambda + \omega - 1)I - \lambda\omega\tilde{L} - \omega\tilde{U}\right) \\
&= \det\left(\sqrt{\lambda\omega^2}\left(\left(\frac{\lambda + \omega - 1}{\sqrt{\lambda\omega^2}}\right) I - \sqrt{\lambda}\tilde{L} - \frac{1}{\sqrt{\lambda}}\tilde{U}\right)\right) \\
&= (\lambda\omega^2)^{n/2} \det\left(\left(\frac{\lambda + \omega - 1}{\sqrt{\lambda\omega^2}}\right) I - \tilde{L} - \tilde{U}\right),
\end{aligned} \tag{6.24}$$

其中最后一个等式由引理 6.7 推得. 令

$$\mu = \frac{\lambda + \omega - 1}{\sqrt{\lambda\omega^2}} \quad 即 \quad (\lambda + \omega - 1)^2 = \lambda\omega^2\mu^2,$$

则由 (6.24) 和性质 (1) 可知 λ 是 G_{SOR} 的特征值当且仅当 μ 是 $G_L = L + U$ 的特征值, 即命题结论成立.

(3) 已经由 (2) 证明. □

推论 6.15 若 A 具有相容次序, 则 $\rho(G_{\text{GS}}) = \rho(G_{\text{J}})^2$, 即当 *Jacobi* 算法收敛时, *G-S* 算法比 *Jacobi* 算法快一倍.

例 6.4 采用红黑排序的二维离散 Poisson 方程, 系数矩阵 $\tilde{T}_{N^2}$ 具有相容次序, 故有 $\rho(G_{\mathrm{GS}}) = \rho(G_{\mathrm{J}})^2$.

下面是关于 SOR 算法的最优参数选取.

定理 6.16 设 A 具有相容次序, G_{J} 的特征值全部为实数, 且 $\rho_J = \rho(G_{\mathrm{J}}) < 1$, 则 SOR 算法的最优参数为

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\rho_J^2}},$$

此时

$$\rho(G_{\mathrm{SOR}}) = \omega_{opt} - 1 = \frac{\rho_J^2}{\left(1+\sqrt{1-\rho_J^2}\right)^2}.$$

进一步, 有

$$\rho(G_{\mathrm{SOR}}) = \begin{cases} \omega - 1, & \omega_{opt} \le \omega \le 2 \\ 1 - \omega + \dfrac{1}{2}\omega^2\rho_J^2 + \omega\rho_J\sqrt{1-\omega+\dfrac{1}{4}\omega^2\rho_J^2}, & 0 < \omega \le \omega_{opt} \end{cases}.$$

证明. 直接求解等式 (6.23), 分情况讨论即可. □

例 6.5 采用红黑排序的二维离散 Poisson 问题的系数矩阵 $\tilde{T}_{N^2}$ 具有相容次序, 且 G_{J} 是对称的, 即 G_{J} 的特征值都是实的. 又由系数矩阵的弱对角占优和不可约性质可知 $\rho(G_{\mathrm{J}}) < 1$, 故上述定理的条件均满足.

6.3.5 M 矩阵与 H 矩阵

参见胡家赣 [47] 和徐树方 [52].

To be continued ...

6.4 加速算法

当迭代解 $x^{(0)}, x^{(1)}, x^{(2)}, \ldots, x^{(k)}$ 已经计算出来后, 我们可以对其进行组合, 得到一个新的近似解, 这样就可以对原算法进行加速.

6.4.1 外推技术

设原迭代格式为

$$x^{(k+1)} = Gx^{(k)} + b. \tag{6.25}$$

由 $x^{(k)}$ 和 $x^{(k+1)}$ 加权组合后可得新的近似解

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega(Gx^{(k)} + b), \tag{6.26}$$

其中 ω 是参数. 这种加速方法就称为 外推算法.

☞ 如果原迭代格式是 Gauss-Seidel 迭代, 虽然 SOR 与外推算法 (6.26) 都是通过加权进行加速, 但格式是不一样的: 前者是在单个分量计算出来后就进行加权, 而后者则是当所有分量都计算出来后才加权.

为了使得迭代格式 (6.26) 尽可能快地收敛, 需要选择 ω 使得其迭代矩阵 $G_\omega \triangleq (1-\omega)I + \omega G$ 的谱半径尽可能地小. 假设 G 的特征值都是实数, 且最大特征值和最小特征值分别为 λ_1 和 λ_n. 于是

$$\rho(G_\omega) = \max_{\lambda\in\sigma(G)} |(1-\omega)+\omega\lambda| = \max\{|1-\omega+\omega\lambda_1|, |1-\omega+\omega\lambda_n|\}.$$

定理 6.17 设 G 的特征值都是实数, 其最大和最小特征值分别为 λ_1 和 λ_n, 且 $1 \notin [\lambda_n, \lambda_1]$, 则

$$\omega_* = \arg\min_{\omega} \rho(G_\omega) = \frac{2}{2-(\lambda_1+\lambda_n)},$$

此时

$$\rho(G_{\omega_*}) = 1 - |\omega_*|d,$$

其中 d 是 1 到 $[\lambda_n, \lambda_1]$ 的距离, 即当 $\lambda_n \le \lambda_1 < 1$ 时, $d = 1-\lambda_1$, 当 $\lambda_1 \ge \lambda_n > 1$ 时, $d = \lambda_n - 1$.

证明. 先考虑 $\lambda_1 < 1$ 的情形. 此时有

$$\max_{\lambda_m \le \lambda \le \lambda_1} |(1-\omega)+\omega\lambda| = \begin{cases} 1-\omega+\lambda_n\omega, & \omega \le 0; \\ 1-\omega+\lambda_1\omega, & 0 < \omega \le \omega_*; \\ -1-\lambda_n\omega+\omega, & \omega > \omega_*. \end{cases}$$

显然, 当 $\omega = \omega_*$ 时取最小值.

对于 $\lambda_n > 1$ 的情形, 可以进行类似的讨论. □

由定理 6.17 可知, $\rho(G_{\omega_*}) = 1 - |\omega_*|d$, 且当 $\omega_* \ne 1$ 时, 外推迭代 (6.26) 比原迭代算法收敛要更快一些.

☞ 最优参数依赖于原迭代矩阵 G 的特征值, 因此实用性不强. 在实际应用时可以估计特征值所在的区间 $[a, b]$, 然后用 a, b 来代替 λ_n 和 λ_1.

JOR 算法

对 Jacobi 迭代进行外推加速, 则可得 JOR (Jacobi over-relaxation) 算法:

$$\begin{aligned} x^{(k+1)} &= (1-\omega)x^{(k)} + \omega(D^{-1}(L+U)x^{(k)} + D^{-1}b) \\ &= x^{(k)} + \omega D^{-1}(b - Ax^{(k)}), \quad k = 0, 1, 2, \ldots. \end{aligned}$$

定理 6.18 设 A 对称正定. 若

$$0 < \omega < \frac{2}{\rho(D^{-1}A)},$$

则 *JOR* 算法收敛.

6.4.2 Chebyshev 加速

本节对外推技巧进行推广.

假定通过迭代格式 (6.25) 已经计算出 $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$, 下面考虑如何将这些近似解进行组合, 以便得到更精确的近似解.

记 $\varepsilon_k = x^{(k)} - x_*$ 为第 k 步迭代解的误差, 则有

$$\varepsilon_k = G\varepsilon_{k-1} = G^2\varepsilon_{k-2} = \cdots = G^k\varepsilon_0.$$

设 $\tilde{x}^{(k)}$ 为 $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$ 的一个线性组合, 即

$$\tilde{x}^{(k)} = \alpha_0 x^{(0)} + \alpha_1 x^{(1)} + \cdots + \alpha_k x^{(k)}, \tag{6.27}$$

其中 α_i 为待定系数, 且满足 $\sum_{i=0}^{k} \alpha_i = 1$. 于是

$$\tilde{x}^{(k)} - x_* = \alpha_0\varepsilon_0 + \alpha_1 G\varepsilon_0 + \cdots + \alpha_k G^k\varepsilon_0 \triangleq p_k(G)\varepsilon_0, \tag{6.28}$$

其中 $p_k(t) = \sum_{i=0}^{k} \alpha_i t^i$ 为 k 次多项式, 且满足 $p_k(1) = 1$.

我们希望通过适当选取参数 α_i, 使得 $\tilde{x}^{(k)} - x_*$ 尽可能地小, 即使得 $\tilde{x}^{(k)}$ 收敛到 x_* 速度远远快于 $x^{(k)}$ 收敛到 x_* 速度. 这种加速方法就称为**多项式加速**或**半迭代方法** (semi-iterative method) .

例 6.6 设 $p_n(t)$ 为 G 的特征多项式, 则 $p_n(G) = 0$, 所以选取 α_i 为 p_n 的系数, 则 $\tilde{x}^{(n)} - x_* = 0$. 但这种选取方法不实用, 原因是:

(1) $p_n(t)$ 的系数并不知道;

(2) 我们通常希望收敛所需的迭代步数 $\ll n$.

下面讨论参数 α_i 的较实用的选取方法. 由 (6.28) 可知

$$\|\tilde{x}^{(k)} - x_*\|_2 = \|p_k(G)\varepsilon_0\|_2 \leq \|p_k(G)\|_2 \cdot \|\varepsilon_0\|_2.$$

因此我们需要求解下面的极小化问题

$$\min_{p\in\mathbb{P}_k,p(1)=1} \|p(G)\|_2. \tag{6.29}$$

一般来说,这个问题是非常困难的. 但在一些特殊情况下,我们可以给出其最优解. 假设迭代矩阵 G 的特征值都是实的,即 G 的 Schur 标准型为

$$G = U(\Lambda + R)U^*,$$

其中 Λ 是对角矩阵,且对角线元素都是实的, R 是严格上三角矩阵, U 是酉矩阵. 于是有

$$\begin{aligned}
\min_{p\in\mathbb{P}_k,p(1)=1} \|p(G)\|_2 &= \min_{p\in\mathbb{P}_k,p(1)=1} \|p(\Lambda+R)\|_2 \\
&= \min_{p\in\mathbb{P}_k,p(1)=1} \|p(\Lambda)\|_2 \\
&= \min_{p\in\mathbb{P}_k,p(1)=1} \max_{1\le i\le n} \left\{|p(\lambda_i)|\right\} \\
&\le \min_{p\in\mathbb{P}_k,p(1)=1} \max_{\lambda\in[\lambda_n,\lambda_1]} \left\{|p(\lambda)|\right\},
\end{aligned} \tag{6.30}$$

其中 λ_1, λ_n 分别表示 G 的最大和最小特征值. 这是带归一化条件的多项式最佳一致逼近问题 (与零的偏差最小). 该问题的解与著名的 Chebyshev 多项式有关.

Chebyshev 多项式

Chebyshev 多项式是一类很重要的正交多项式, 在函数逼近, 函数插值, 数值积分等方面都有着重要的应用.

Chebyshev 多项式 $T_k(t)$ 可以通过下面的递归方式来定义:

$$\begin{aligned}
&T_0(t) = 1, \quad T_1(t) = t, \\
&T_k(t) = 2\,t\,T_{k-1}(t) - T_{k-2}(t),\ k = 2, 3, \dots,
\end{aligned} \tag{6.31}$$

也可以直接由下面的式子定义

$$\begin{aligned}
T_k(t) &= \frac{1}{2}\left[\left(t+\sqrt{t^2-1}\right)^k + \left(t+\sqrt{t^2-1}\right)^{-k}\right] \\
&= \frac{1}{2}\left[\left(t+\sqrt{t^2-1}\right)^k + \left(t-\sqrt{t^2-1}\right)^{k}\right],
\end{aligned}$$

或者

$$T_k(t) = \begin{cases} \cos(k\arccos t), & |t|\le 1 \\ \cosh(k\,\mathrm{arccosh}\,t), & |t|>1 \end{cases},$$

其中 cosh 为双曲余弦, 即 $\cosh(t) = \dfrac{e^t+e^{-t}}{2}$. 容易验证, Chebyshev 多项式具有下面的带权正交性质

$$\int_{-1}^{1} \frac{T_k(t)T_l(t)}{\sqrt{1-t^2}}\,\mathrm{d}t = \begin{cases} \pi, & k=l=0; \\ \dfrac{\pi}{2}, & k=l>0; \\ 0, & k\ne l. \end{cases}$$

事实上, 在线性空间 $C[-1,1]$ 上定义带权内积

$$(f,g)_w = \int_{-1}^{1} w(x)f(x)g(x)\,\mathrm{d}x,$$

其中 $w(x)$ 是 $[-1,1]$ 上的权函数, 即满足:

(1) $w(x) \geq 0, \forall\, x \in [-1,1]$;

(2) $\int_{-1}^{1} x^k w(x)\,\mathrm{d}x$ 存在, 其中 k 为任意非负整数;

(3) 若 $\int_{-1}^{1} g(x)w(x)\,\mathrm{d}x = 0$, 则 $g(x) = 0$, 其中 $g(x)$ 是 $[-1,1]$ 上的任意非负连续函数.

☞ 以上关于权函数的定义可以推广到一般区间 $[a,b]$ 上.

在该内积下, 将线性无关函数组 $\{1, x, x^2, \ldots, x^n, \ldots\}$ 正交化后得到的正交多项式就是 Chebyshev 多项式.

引理 6.9 *Chebyshev 多项式具有下面的性质:*

(1) $T_k(1) = 1$;

(2) $T_k(t)$ 的首项系数为 2^{k-1};

(3) $T_k(-t) = (-1)^k T_k(t)$, 即 $T_{2k}(t)$ 只含偶次项, $T_{2k+1}(t)$ 只含奇次项;

(4) 当 $|t| \leq 1$ 时 $|T_k(t)| \leq 1$; 当 $|t| > 1$ 时 $|T_k(t)| > 1$;

(5) $T_k(t) = 0$ 的解为 $t_i = \cos\dfrac{(2i-1)\pi}{2k}$, $i = 1, \ldots, k$;

(6) $T_k(t)$ 有 $n-1$ 个极值点: $t_i = \cos\dfrac{k\pi}{n}$, $i = 1, 2, \ldots, k-1$;

(7) $T_k(1+\epsilon) \geq \dfrac{1+k\sqrt{2\epsilon}}{2}, \quad \forall \epsilon > 0.$

下面的结论表明, 在所有首项系数为 1 的 k 次多项式中, $p_k(t) = \dfrac{1}{2^{k-1}}T_k(t)$ 在 $[-1,1]$ 上与零的偏差是最小的 (在无穷范数意义下).

定理 6.19 设 $p_k(t) = \dfrac{1}{2^{k-1}}T_k(t)$, 则

$$\max_{-1\leq t\leq 1} |p_k(t)| \leq \max_{-1\leq t\leq 1} |p(t)|, \quad \forall\, p(t) \in \tilde{\mathbb{P}}_k,$$

其中 $\tilde{\mathbb{P}}_k$ 表示所有首项系数为 1 的 k 次多形式组成的集合, 即

$$\|p_k(t)\|_\infty = \min_{p(t)\in\tilde{\mathbb{P}}_k} \|p(t)\|_\infty.$$

这里的范数 $\|\cdot\|_\infty$ 是 $C[-1,1]$ 上的无穷范数, 即 $\|f\|_\infty = \max_{x\in[-1,1]} |f(x)|$, $f(x) \in C[-1,1]$.

该性质可用于计算首项系数非零的 n 次多项式在 $[-1,1]$ 上的 $n-1$ 次最佳一致逼近多项式. 利用这个性质, 我们可以采用 Chebyshev 多项式的零点作为节点进行多项式插值, 以使得插值的总体误差达到最小化.

Chebyshev 另外一个重要性质是下面的最小最大性质.

定理 6.20 设 $\eta \in \mathbb{R}$ 满足 $|\eta| > 1$, 则下面的最小最大问题

$$\min_{p(t)\in\mathbb{P}_k, p(\eta)=1} \max_{-1\le t\le 1} |p(t)|$$

的唯一解为

$$\tilde{T}_k(t) \triangleq \frac{T_k(t)}{T_k(\eta)}.$$

通过简单的仿射变换, 该定理的结论可以推广到一般区间.

定理 6.21 设 $\alpha, \beta, \eta \in \mathbb{R}$ 满足 $\alpha < \beta$ 且 $|\eta| \notin [\alpha, \beta]$. 则下面的最小最大问题

$$\min_{p(t)\in\mathbb{P}_k, p(\eta)=1} \max_{\alpha\le x\le \beta} |p(t)|$$

的唯一解为

$$\hat{T}_k(t) \triangleq \frac{T_k\left(\dfrac{2t-(\beta+\alpha)}{\beta-\alpha}\right)}{T_k\left(\dfrac{2\eta-(\beta+\alpha)}{\beta-\alpha}\right)}.$$

Chebyshev 加速算法

考虑迭代格式 (6.25), 我们假定:

(1) 迭代矩阵 G 的特征值都是实数;

(2) 迭代矩阵谱半径 $\rho = \rho(G) < 1$, 故 $\lambda(G) \in [-\rho, \rho] \subset (-1, 1)$.

于是最小最大问题 (6.30) 就转化为

$$\min_{p\in\mathbb{P}_k, p(1)=1} \max_{\lambda\in[-\rho,\rho]} \{|p(\lambda)|\}.$$

由于 $1 \neq [-\rho, \rho]$, 根据定理 6.21, 上述问题的解为

$$p_k(t) = \frac{T_k(t/\rho)}{T_k(1/\rho)}.$$

下面考虑 $\tilde{x}^{(k)}$ 的计算. 我们无需先计算出 $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$, 然后再通过线性组合 (6.27) 来计算 $\tilde{x}^{(k)}$. 事实上, 我们可以通过 Chebyshev 多项式的三项递推公式 (6.31), 由 $\tilde{x}^{(k-1)}$ 和 $\tilde{x}^{(k-2)}$ 直接计算出 $\tilde{x}^{(k)}$. 这样做的另一个好处是无需存储所有的 $\tilde{x}^{(i)}$. 下面给出具体的推导公式.

令 $\mu_k = \dfrac{1}{T_k(1/\rho)}$, 即 $T_k(1/\rho) = \dfrac{1}{\mu_k}$. 由三项递推公式 (6.31) 可得

$$\frac{1}{\mu_k} = \frac{2}{\rho} \cdot \frac{1}{\mu_{k-1}} - \frac{1}{\mu_{k-2}}.$$

所以

$$\tilde{x}^{(k)} - x_* = p_k(G)\,\varepsilon_0 = \mu_k T_k(G/\rho)\,\varepsilon_0$$

$$
\begin{aligned}
&= \mu_k \left[\frac{2G}{\rho} \cdot T_{k-1}(G/\rho) - T_{k-2}(G/\rho) \right] \varepsilon_0 \\
&= \mu_k \left[\frac{2G}{\rho} \cdot \frac{1}{\mu_{k-1}} p_{k-1}(G/\rho)\varepsilon_0 - \frac{1}{\mu_{k-2}} p_{k-2}(G/\rho)\varepsilon_0 \right] \\
&= \mu_k \left[\frac{2G}{\rho} \cdot \frac{1}{\mu_{k-1}} (\tilde{x}^{(k-1)} - x_*) - \frac{1}{\mu_{k-2}} (\tilde{x}^{(k-2)} - x_*) \right].
\end{aligned}
$$

整理后可得

$$
\tilde{x}^{(k)} = \frac{2\mu_k}{\mu_{k-1}} \cdot \frac{G}{\rho} \tilde{x}^{(k-1)} - \frac{\mu_k}{\mu_{k-2}} \tilde{x}^{(k-2)} + d_k,
$$

其中

$$
\begin{aligned}
d_k &= x_* - \frac{2\mu_k}{\mu_{k-1}} \cdot \frac{G}{\rho} x_* + \frac{\mu_k}{\mu_{k-2}} x_* \\
&= x^* - \frac{2\mu_k}{\mu_{k-1}} \cdot \frac{x_* - g}{\rho} + \frac{\mu_k}{\mu_{k-2}} x_* \\
&= \mu_k \left(\frac{1}{\mu_k} - \frac{2}{\rho \mu_{k-1}} + \frac{1}{\mu_{k-2}} \right) x_* + \frac{2\mu_k g}{\mu_{k-1}\rho} \\
&= \frac{2\mu_k g}{\mu_{k-1}\rho}.
\end{aligned}
$$

由此, 我们可以得到迭代格式 (6.25) 的 Chebyshev 加速算法.

算法 6.8 Chebyshev 加速算法

1: Set $\mu_0 = 1, \mu_1 = \rho = \rho(G), \tilde{x}^{(0)} = x^{(0)}, k = 1$
2: compute $\tilde{x}^{(1)} = Gx^{(0)} + g$
3: **while** not converge **do**
4: $\quad k = k + 1$
5: $\quad \mu_k = \left(\frac{2}{\rho} \cdot \frac{1}{\mu_{k-1}} - \frac{1}{\mu_{k-2}} \right)^{-1}$
6: $\quad \tilde{x}^{(k)} = \frac{2\mu_k}{\mu_{k-1}} \cdot \frac{G}{\rho} \tilde{x}^{(k-1)} - \frac{\mu_k}{\mu_{k-2}} \tilde{x}^{(k-2)} + \frac{2\mu_k}{\mu_{k-1}\rho} \cdot g$
7: **end while**

☞ 该算法的每步迭代中只有一次矩阵向量乘积, 故算法每个迭代步的整体运算量与原迭代格式的每个迭代步的运算量基本相当.

☞ 设 $\lambda(G) \in [\alpha, \beta]$, 且 $-1 < \alpha \leq \beta < 1$, 则我们也可以构造出相应的 Chebyshev 加速算法.

例 6.7 针对二维离散 Poisson 方程, 比较 Jacobi, JOR 和 Chebyshev 加速算法的收敛性质. (绘制收敛曲线)

To be continued ...

SSOR 算法的 Chebyshev 加速

SSOR 迭代矩阵为

$$
G_{\text{SSOR}} = (D - \omega U)^{-1} \big[(1-\omega)D + \omega L \big] (D - \omega L)^{-1} \big[(1-\omega)D + \omega U \big].
$$

当 A 对称时, 有 $L = U^T$, 故

$$
\begin{aligned}
& (D-\omega U) G_{\text{SSOR}} (D-\omega U)^{-1} \\
& = \big[(1-\omega)D + \omega L\big](D-\omega L)^{-1}\big[(1-\omega)D + \omega L^T\big](D-\omega L^T)^{-1} \\
& = \big[(2-\omega)D(D-\omega L)^{-1} - I\big]\big[(2-\omega)D(D-\omega L^T)^{-1} - I\big] \\
& = I - (2-\omega)D\big[(D-\omega L)^{-1} + (D-\omega L^T)^{-1}\big] + (2-\omega)^2 D(D-\omega L)^{-1}D(I-\omega L^T)^{-1}.
\end{aligned}
$$

假定 D 的对角线元素全是正的, 则

$$
\begin{aligned}
& D^{-1/2}(D-\omega U) G_{\text{SSOR}} (D-\omega U)^{-1} D^{1/2} \\
& = I - (2-\omega)D^{-1/2}\big[(D-\omega L)^{-1} + (D-\omega L^T)^{-1}\big]D^{1/2} \\
& \quad + (2-\omega)^2 D^{-1/2}(D-\omega L)^{-1}D(I-\omega L^T)^{-1}D^{1/2}.
\end{aligned}
$$

这是一个对称矩阵, 故 G_{SSOR} 具有实特征值. 所以我们可以对其实行 Chebyshev 加速. 但我们需要估计 G_{SSOR} 的谱半径.

☞ 若存在矩阵 W 使得 $W^{-1}AW$ 是对称矩阵, 则称 A 是可对称化的, 即 A 相似于一个对称矩阵.

6.5 快速 Poisson 算法

如果已经知道矩阵 A 的特征值分解 $A = X\Lambda X^{-1}$, 则 $Ax = b$ 的解可表示为

$$x = A^{-1}b = X\Lambda^{-1}X^{-1}b.$$

如果 A 是正规矩阵, 即 X 是酉矩阵, 则

$$x = A^{-1}b = X\Lambda^{-1}X^{*}b.$$

一般来说, 我们不会采用这种特征值分解的方法来解线性方程组, 因为计算特征值分解通常比解线性方程组更困难. 但在某些特殊情况下, 我们可以由此得到快速算法.

考虑二维离散 Poisson 方程

$$Tu = h^2 f, \tag{6.32}$$

其中

$$T = I \otimes T_n + T_n \otimes I, \quad T_n = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{n\times n}.$$

由定理 6.1 可知

$$T = (Z \otimes Z)(I \otimes \Lambda + \Lambda \otimes I)(Z \otimes Z)^T,$$

其中 $Z = [z_1, z_2, \dots, z_n]$ 是正交矩阵. 这里

$$z_k = \sqrt{\frac{2}{n+1}} \cdot \left[\sin\frac{k\pi}{n+1}, \sin\frac{2k\pi}{n+1}, \dots, \sin\frac{nk\pi}{n+1}\right]^T, \quad k = 1, 2, \dots, n.$$

所以, 方程 (6.32) 的解为

$$u = T^{-1}h^2 f = [(Z \otimes Z)(I \otimes \Lambda + \Lambda \otimes I)^{-1}(Z \otimes Z)^T]\, h^2 f.$$

因此, 主要的运算是 $Z \otimes Z$ 与向量的乘积, 以及 $(Z \otimes Z)^T$ 与向量的乘积. 而这些乘积可以通过快速 Fourier 变换 (FFT) 来实现.

6.5.1 FFT

FFT (快速 Fourier 变换) 用来计算离散 Fourier 变换 (DFT) 矩阵与向量乘积的一种快速算法.

设 $x = [x_0, x_1 \dots, x_{n-1}]^T \in \mathbb{C}^n$, 其 DFT 定义为 $y = \text{DFT}(x) = [y_0, y_1 \dots, y_{n-1}]^T \in \mathbb{C}^n$, 其中

$$y_k = \sum_{j=0}^{n-1} \omega_n^{kj} x_j, \quad k = 0, 1, \dots, n-1.$$

这里

$$\omega_n = e^{\frac{-2\pi \mathbf{i}}{n}} = \cos(2\pi/n) - \mathbf{i}\sin(2\pi/n)$$

是 1 的一个 n 次本原根 (primitive n-th root of unity), $\mathbf{i}$ 是虚部单位.

☞ 这里说 ω_n 是 primitive n-th root of unity 是指 $\omega_n^n = 1$ 且

$$\omega_n^k \neq 1, \quad k = 1, 2, \dots, n-1.$$

构造矩阵 $F_n = [f_{kj}] \in \mathbb{C}^{n\times n}$, 其中 $f_{kj} = \omega_n^{kj} = e^{\frac{-2kj\pi \mathbf{i}}{n}} = \cos(2kj\pi/n) - \mathbf{i}\sin(2kj\pi/n)$, 即

$$F_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)^2} \end{bmatrix},$$

则有

$$y = \mathrm{DFT}(\mathrm{x}) = F_n x.$$

我们称矩阵 F_n 为 DFT 矩阵. 易知 DFT 矩阵具有以下性质:

(1) F_n 是对称矩阵 (但不是 Hermite 对称);

(2) $F_n^* F_n = nI$, 所以 $\frac{1}{\sqrt{n}} F_n$ 是酉矩阵.

相应的离散 Fourier 反变换定义为 $x = \mathrm{IDFT}(y)$, 其中

$$x_j = \frac{1}{n}\sum_{k=0}^{n-1} \omega_n^{-jk} y_k, \quad j = 0, 1, \ldots, n-1.$$

写成矩阵形式为

$$x = \frac{1}{n} F^* y.$$

DFT 和 IDFT 满足下面的性质:

$$\begin{aligned} \mathrm{IDFT}(\mathrm{DFT}(x)) &= x, \\ \mathrm{DFT}(\mathrm{IDFT}(y)) &= y. \end{aligned}$$

☞ 在 MATLAB 中, 计算 DFT 和 IDFT 的函数分别为 `fft` 和 `ifft`, 即: `y=fft(x)`, `x=ifft(y)`. (测试代码见 `FFT_test.m`)

6.5.2 离散 Sine 变换

离散 Sine 变换有多种定义, 我们这里只介绍与求解 Poission 方程有关的一种定义, 其它定义请参考 [32].

设 $x = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$, 其 **离散 Sine 变换 (DST)** 定义为 $y = \mathrm{DST}(x) = [y_1, y_2, \ldots, y_n]^T \in \mathbb{R}^n$, 其中

$$y_k = \sum_{j=1}^{n} x_j \sin\left(\frac{kj\pi}{n+1}\right), \quad k = 1, 2, \ldots, n.$$

对应的离散 Sine 反变换记为 IDST, 即 $x = \mathrm{IDST}(\mathrm{y})$, 其中

$$x_j = \frac{2}{n+1}\sum_{k=1}^{n} y_k \sin\left(\frac{jk\pi}{n+1}\right), \quad j = 1, 2, \ldots, n.$$

DST 和 IDST 满足下面的性质:

$$\begin{aligned} \mathrm{IDST}(\mathrm{DST}(x)) &= x, \\ \mathrm{DST}(\mathrm{IDST}(y)) &= y. \end{aligned}$$

☞ 在 MATLAB 中, 计算 DST 和 IDST 的函数分别为 `dst` 和 `idst`, 即: `y=dst(x)`, `x=idst(y)`. (测试代码见 `DST_test.m`)

6.5.3 Possion 方程与 DST

我们首先考虑矩阵 Z 与一个任意给定向量 b 的乘积. 设 $y = Zb$, 则

$$y_k = \sum_{j=1}^{n} Z(k,j) b_j = \sqrt{\frac{2}{n+1}} \sum_{j=1}^{n} b_j \sin\left(\frac{kj\pi}{n+1}\right) = \sqrt{\frac{2}{n+1}} \cdot \mathrm{DST}(b).$$

因此, 乘积 $y = Zb$ 可以通过 DST 来实现. 类似地, 乘积 $y = Z^T b = Z^{-1} b$ 可以通过离散 Sine 反变换 IDST 实现, 即

$$y = Z^T b = Z^{-1} b = \left(\sqrt{\frac{2}{n+1}}\right)^{-1} \mathrm{IDST}(b).$$

所以对于一维离散 Poisson 方程, 其解为

$$u = T_n^{-1}(h^2 f) = (Z\Lambda^{-1}Z^T)(h^2 f) = h^2 Z\Lambda^{-1}Z^T f = h^2 \cdot \mathrm{DST}(\Lambda^{-1}\,\mathrm{IDST}(b)).$$

而对于二维离散 Poisson 方程, 我们需要计算 $(Z \otimes Z)b$ 和 $(Z^T \otimes Z^T)b$. 它们对应的是二维离散 Sine 变换和二维离散 Sine 反变换.

设 $b = [b_1^T, b_2^T, \ldots, b_n^T]^T \in \mathbb{R}^{n^2}$, 其中 $b_k \in \mathbb{R}^{n\times n}$. 令 $B = [b_1, b_2, \ldots, b_n] \in \mathbb{R}^{n\times n}$, 则由 Kronecker 乘积的性质可知

$$(Z \otimes Z)b = (Z \otimes Z)\mathrm{vec}(B) = \mathrm{vec}(ZBZ^T) = \mathrm{vec}\left((Z(ZB)^T)^T\right).$$

因此, 我们仍然可以使用 DST 来计算 $(Z \otimes Z)b$. 类似地, 我们可以使用 IDST 来计算 $(Z^T \otimes Z^T)b$.

算法 6.9 二维离散 Poisson 方程的快速算法

1: 计算 $b = h^2 f$
2: $B = \mathrm{reshape}(b, n, n)$
3: $B_1 = (Z^T B)^T = (\mathrm{IDST}(B))^T$
4: $B_2 = (Z^T B_1)^T = (\mathrm{IDST}(B_1))^T$
5: $b_1 = (I \otimes \Lambda + \Lambda \otimes I)^{-1}\mathrm{vec}(B_2)$
6: $B_3 = \mathrm{reshape}(b_1, n, n)$
7: $B_4 = (ZB_3)^T = (\mathrm{DST}(B_3))^T$
8: $B_5 = (ZB_4)^T = (\mathrm{DST}(B_4))^T$
9: $u = \mathrm{reshape}(B_5, n^2, 1)$

MATLAB 程序见 `Poisson_DST.m`

6.6 交替方向与 HSS 方法

6.6.1 多步迭代法

设 $A = M_1 - N_1 = M_2 - N_2$ 是 A 的两个矩阵分裂, 则可以构造迭代格式

$$\begin{cases} M_1 x^{(k+\frac{1}{2})} = N_1 x^{(k)} + b, \\ M_2 x^{(k+1)} = N_2 x^{(k+\frac{1}{2})} + b, \end{cases} \qquad k = 0, 1, 2, \ldots. \tag{6.33}$$

这就是两步迭代算法, 对应的分裂称为二重分裂. 易知, 两步迭代格式 (6.33) 的迭代矩阵为

$$G = M_2^{-1} N_2 M_1^{-1} N_1.$$

因此, 其收敛的充要条件是 $\rho(M_2^{-1} N_2 M_1^{-1} N_1) < 1$.

类似地, 我们可以推广到多步迭代算法. 设 l 是一个正整数, 则 A 的 l 重分裂为

$$A = M_1 - N_1 = M_2 - N_2 = \cdots = M_l - N_l,$$

相应的多步迭代算法为

$$\begin{cases} M_1 x^{(k+\frac{1}{l})} = N_1 x^{(k)} + b, \\ M_2 x^{(k+\frac{2}{l})} = N_2 x^{(k+\frac{1}{l})} + b, \\ \cdots \\ M_l x^{(k+1)} = N_l x^{(k+\frac{l-1}{l})} + b, \end{cases} \qquad k = 0, 1, 2, \ldots.$$

6.6.2 交替方向法

交替方向法 (alternating direction implicit, ADI) 是由 Peaceman 和 Rachford [30] 于 1955 年提出, 用于计算偏微分方程的数值解, 因此也称为 PR 算法. 其本质上也可以看成是一个二重迭代算法.

设 $A = A_1 + A_2$, 则 ADI 迭代格式为

$$\begin{cases} (\alpha I + A_1) x^{(k+\frac{1}{2})} = (\alpha I - A_2) x^{(k)} + b, \\ (\alpha I + A_2) x^{(k+1)} = (\alpha I - A_1) x^{(k+\frac{1}{2})} + b, \end{cases} \qquad k = 0, 1, 2, \ldots, \tag{6.34}$$

其中 $\alpha \in \mathbb{R}$ 是迭代参数. 易知 ADI 算法的迭代矩阵为

$$G_{\text{ADI}} = (\alpha I + A_2)^{-1} (\alpha I - A_1) (\alpha I + A_1)^{-1} (\alpha I - A_2).$$

它相似于

$$\tilde{G} \triangleq (\alpha I - A_1)(\alpha I + A_1)^{-1} (\alpha I - A_2)(\alpha I + A_2)^{-1}.$$

所以 ADI 迭代 (6.35) 收敛的充要条件是

$$\rho(\tilde{G}) < 1.$$

若 A 对称正定, 且 A_1 和 A_2 中有一个是对称正定, 另一个是对称半正定, 则有下面的收敛定理.

定理 6.22 设 $A \in \mathbb{R}^{n\times n}$ 对称正定, $A = A_1 + A_2$, 其中 A_1 和 A_2 中有一个是对称正定, 另一个是对称半正定, 则对任意正数 $\alpha > 0$, 有 $\rho(\tilde{G}) < 1$, 即 *ADI* 迭代算法 (6.35) 收敛.

证明. 不妨假设 A_1 对称正定, A_2 对称半正定. 则

$$\|(\alpha I - A_1)(\alpha I + A_1)^{-1}\|_2 = \max_{\lambda\in\sigma(A_1)} \left|\frac{\alpha-\lambda}{\alpha+\lambda}\right| < 1,$$
$$\|(\alpha I - A_2)(\alpha I + A_2)^{-1}\|_2 = \max_{\lambda\in\sigma(A_2)} \left|\frac{\alpha-\lambda}{\alpha+\lambda}\right| \leq 1.$$

所以,

$$\rho(\tilde{G}) \leq \|\tilde{G}\|_2 \leq \|(\alpha I - A_1)(\alpha I + A_1)^{-1}\|_2 \cdot \|(\alpha I - A_2)(\alpha I + A_2)^{-1}\|_2 < 1.$$

□

6.6.3 HSS 方法

HSS 方法全称为 Hermitian and Skew-Hermitian Splitting method, 是由 Bai, Golub 和 Ng [4] 于 2003 年提出.

设 $A = H + S$, 其中 H 和 S 分别是 A 的对称与反对称 (斜对称, Skew-Hermitian) 部分, 即

$$H = \frac{A + A^T}{2}, \quad S = \frac{A - A^T}{2}.$$

该分裂就称为 HS 分裂, 即 HSS.

☞ 任何一个矩阵都具有 HS 分裂. 如果 $A \in \mathbb{C}^{n\times n}$, 则取共轭转置.

类似于 ADI 方法, 我们可得下面的 HSS 方法

$$\begin{cases} (\alpha I + H)x^{(k+\frac{1}{2})} = (\alpha I - S)x^{(k)} + b, \\ (\alpha I + S)x^{(k+1)} = (\alpha I - H)x^{(k+\frac{1}{2})} + b, \end{cases} \quad k = 0, 1, 2, \ldots. \tag{6.35}$$

易知, HSS 方法的迭代矩阵为

$$G_{\text{HSS}} = (\alpha I + S)^{-1}(\alpha I - H)(\alpha I + H)^{-1}(\alpha I - S).$$

同样, 它相似于

$$\tilde{G} \triangleq (\alpha I - H)(\alpha I + H)^{-1}(\alpha I - S)(\alpha I + S)^{-1}.$$

我们首先考察矩阵 $(\alpha I - S)(\alpha I + S)^{-1}$. 事实上, 它是一个酉矩阵 (见习题 (5.5)). 因此

$$\|(\alpha I - S)(\alpha I + S)^{-1}\|_2 = 1.$$

由于 H 是对称矩阵, 因此

$$\|(\alpha I - H)(\alpha I + H)^{-1}\|_2 = \max_{\lambda\in\sigma(H)} \left|\frac{\alpha-\lambda}{\alpha+\lambda}\right|.$$

定理 6.23 设 $A \in \mathbb{R}^{n\times n}$ 正定, 则对任意正数 $\alpha > 0$, 有 $\rho(\tilde{G}) < 1$, 即 *HSS* 迭代算法 (??) 收敛.

证明. 由于 A 正定, 即 H 对称正定, 故

$$\|(\alpha I - H)(\alpha I + H)^{-1}\|_2 = \max_{\lambda\in\sigma(H)} \left|\frac{\alpha-\lambda}{\alpha+\lambda}\right| < 1.$$

所以, 结论成立. □

参数 α 的选取

HSS 方法的推广

PSS, NSS, AHSS, MHSS 等

6.7 多重网格方法

参见李大明 [49].

To be continued ...

6.8 Krylov 子空间迭代算法

子空间迭代算法的基本思想是在一个维数较低的子空间中寻找解析解的一个 “最佳” 逼近. 子空间迭代算法的主要过程可以分为下面三步:

(1) 寻找合适的子空间;

(2) 在该子空间中求 “最佳” 近似解;

(3) 若这个近似解不满足精度要求, 则重新构造一个新的子空间, 然后返回第 (1) 步.

其中第 (1) 步的第 (3) 步是关于子空间的选取和更新问题. 目前成功的解决方案就是使用 Krylov 子空间.

6.8.1 Krylov 子空间

设 $A \in \mathbb{R}^{n\times n}$, $r \in \mathbb{R}^n$, 则由 A 和 r 生成的 Krylov 子空间为

$$\mathcal{K}_m(A,r) = \mathrm{span}\{r, Ar, A^2r, \dots, A^{m-1}r\}, \quad m \le n.$$

通常简记为 $\mathcal{K}_m$.

下面我们需要在 $\mathcal{K}_m$ 中寻找解析解的 “最佳” 近似值 $x^{(m)}$. 假设 $r, Ar, A^2r, \dots, A^{m-1}r$ 线性无关, 则 $\dim \mathcal{K}_m = m$. 若 $v_1, v_2, \dots, v_m$ 是 $\mathcal{K}_m$ 的一组基, 则 $\mathcal{K}_m$ 中的任意向量 x 均可表示为

$$x = y_1v_1 + y_2v_2 + \cdots + y_mv_m = V_my,$$

其中 $y = [y_1, y_2, \dots, y_m]^T$ 为线性表出系数, $V_m = [v_1, v_2, \dots, v_m]$. 于是, 计算 “最佳” 近似 $x^{(m)}$ 就转化为寻找一组合适的基 $v_1, v_2, \dots, v_m$ 和 $x^{(m)}$ 在这组基下面的表出系数 $y^{(m)}$.

Arnoldi 过程

首先考虑基的选取. 由于 $r, Ar, A^2r, \dots, A^{m-1}r$ 线性无关, 因此它们就组成 $\mathcal{K}_m$ 的一组基. 但为了确保算法的稳定性, 一般来说, 我们通常希望选取一组标准正交基. 我们可以通过对向量组 $\{r, Ar, A^2r, \dots, A^{m-1}r\}$ 进行 Gram-Schmidt 正交化过程得到一组标准正交基, 这个过程就称为 Arnoldi 过程.

算法 6.10 Arnoldi 过程 (MGS)

1: $v_1 = r/\|r\|_2$
2: **for** $j = 1$ to m **do**
3: $\quad z = Av_j$
4: $\quad$ **for** $i = 1$ to j **do**
5: $\quad\quad h_{i,j} = (v_i, z)$ % 内积
6: $\quad\quad z = z - h_{i,j}v_i$
7: $\quad$ **end for**
8: $\quad h_{j+1,j} = \|z\|_2$
9: $\quad$ **if** $h_{j+1,j} = 0$ **then**
10: $\quad\quad$ break
11: $\quad$ **end if**
12: $\quad v_{j+1} = z/h_{j+1,j}$

13: **end for**

记 $V_m = [v_1, v_2, \dots, v_m]$,

$$H_{m+1,m} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,m} \\ & h_{3,2} & h_{3,3} & \cdots & h_{3,m} \\ & & \ddots & \ddots & \vdots \\ & & & h_{m,m-1} & h_{m,m} \\ & & & & h_{m+1,m} \end{bmatrix} \in \mathbb{R}^{(m+1)\times m},$$

则由 Arnoldi 过程可知

$$h_{j+1,j}v_{j+1} = Av_j - h_{1,j}v_1 - h_{2,j}v_2 - \cdots - h_{j,j}v_j,$$

即

$$Av_j = \sum_{i=1}^{j+1} h_{i,j}v_i = V_{m+1} \begin{bmatrix} h_{1,j} \\ \vdots \\ h_{j+1,j} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = V_{m+1}H_{m+1,m}(:,j).$$

所以有

$$AV_m = V_{m+1}H_{m+1,m} = V_mH_m + h_{m+1,m}v_{m+1}e_m^T, \tag{6.36}$$

其中 $H_m = H_{m+1,m}(1:m, 1:m)$, $e_m = [0,\dots,0,1]^T \in \mathbb{R}^m$. 由于 V_m 是列正交矩阵, 上式两边同乘 V_m^T 可得

$$V_m^TAV_m = H_m. \tag{6.37}$$

等式 (6.36) 和 (6.37) 是 Arnoldi 过程的两个重要性质.

由于 $H_{m+1,m}$ 是上 Hessenberg 矩阵, 因此 Arnoldi 过程也称为部分上 Hessenberg 化过程.

Lanczos 过程

如果 A 是对称矩阵, 则 H_m 为对称三对角矩阵, 此时将其记为 T_m, 即

$$T_m = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \beta_{m-1} & \alpha_m \end{bmatrix}. \tag{6.38}$$

与 Arnoldi 过程类似, 我们有下面的性质

$$AV_m = V_mT_m + \beta_mv_{m+1}e_m^T, \tag{6.39}$$

$$V_m^TAV_m = T_m. \tag{6.40}$$

考察关系式 (6.39) 两边的第 j 列可知

$$\beta_j v_{j+1} = Av_j - \alpha_j v_j - \beta_{j-1} v_{j-1}, \quad j = 1, 2, \ldots,$$

这里我们令 $v_0 = 0$ 和 $\beta_0 = 0$. 根据这个三项递推公式, Arnoldi 过程可简化为下面的 Lanczos 过程.

算法 6.11 Lanczos 过程

1: Set $v_0 = 0$ and $\beta_0 = 0$
2: $v_1 = r/\|r\|_2$
3: **for** $j = 1$ to m **do**
4: $\quad z = Av_j$
5: $\quad \alpha_j = (v_j, z)$
6: $\quad z = z - \alpha_j v_j - \beta_{j-1} v_{j-1}$
7: $\quad \beta_j = \|z\|_2$
8: $\quad$ **if** $\beta_j = 0$ **then**
9: $\quad\quad$ break
10: $\quad$ **end if**
11: $\quad v_{j+1} = z/\beta_j$
12: **end for**

6.8.2 Krylov 子空间迭代算法一般格式

Krylov 子空间迭代算法的一般过程为:

(1) 令 $m = 1$
(2) 定义 Krylov 子空间 $\mathcal{K}_m$;
(3) 在该子空间中求出 "最佳" 近似解; (事实上, 是在仿射空间 $x^{(0)} + \mathcal{K}_m$ 中寻找 "最佳" 近似解)
(4) 如果这个近似解满足精度要求, 则迭代结束; 否则令 $m \leftarrow m + 1$, 即将 Krylov 子空间的维数增加一维, 并返回到第 (3) 步.

算法 6.12 Krylov 子空间迭代算法

1: Choose an initial guess $x^{(0)}$
2: computet $r_0 = b - Ax^{(0)}$ and $v_1 = r_0/\|r_0\|_2$
3: find the "best" approximate solution $x^{(1)} \in x^{(0)} + \mathcal{K}_1 = x^{(0)} + \text{span}\{v_1\}$
4: **if** $x^{(1)}$ is ok **then** quit
5: **for** $m = 2$ to n **do**
6: $\quad$ compute v_m with Arnoldi or Lanczos process
7: $\quad$ find the "best" approximate solution $x^{(m)} \in x^{(0)} + \mathcal{K}_m = x^{(0)} + \text{span}\{v_1, \ldots, v_m\}$
8: $\quad$ **if** $x^{(m)}$ is ok **then** quit
9: **end for**

子空间的选取和更新问题可以通过 Krylov 子空间来解决. 下面需要考虑如何寻找方程组在仿射空间 $x^{(0)} + \mathcal{K}_m$ 中的 "最佳" 近似解 $x^{(m)}$. 首先, 我们必须给出 "最佳" 的定义, 即 $x^{(m)}$ 满足什

么条件时才是“最佳”的, 不同的定义会导致不同的算法.

我们很自然地想到用近似解与真解之间的距离来衡量“最佳”, 即使得 $x^{(m)} - x_*$ 在某种意义下最小, 如 $\|x^{(m)} - x_*\|_2$ 达到最小. 但是由于 x_* 不知道, 因此这种方式往往是不实用.

实用的“最佳”方式有:

(1) 满足 $\|r_m\|_2 = \|b - Ax^{(m)}\|_2$ 最小, 即极小化残量 $r_m = b - Ax^{(m)}$. 这种方式是可行的, 当 A 对称时, 相应的算法即为 MINRES 算法, 当 A 不对称时, 相应的算法即为 GMRES 算法;

(2) 要求残量与 $\mathcal{K}_m$ 正交, 即 $r_m \perp \mathcal{K}_m$. 这个正交性条件称为 Galerkin 条件. 当 A 对称时, 相应的算法即为 SYMMLQ 算法, 当 A 对称正定时, 相应的算法即为 CG 算法, 当 A 不对称时, 相应的算法即为 FOM 算法;

(3) 若 A 对称正定, 我们也可以极小化残量的能量范数 $\|r_m\|_{A^{-1}}$, 其中

$$\begin{aligned}\|r_m\|_{A^{-1}} \triangleq \left(r_m^T A^{-1} r_m\right)^{\frac{1}{2}} &= \left((b - Ax^{(m)})^T A^{-1}(b - Ax^{(m)})\right)^{\frac{1}{2}} \\ &= \left((A^{-1}b - x^{(m)})^T A(A^{-1}b - x^{(m)})\right)^{\frac{1}{2}} \\ &= \left((x_* - x^{(m)})^T A(x_* - x^{(m)})\right)^{\frac{1}{2}} \\ &= \|x_* - x^{(m)}\|_A.\end{aligned}$$

下面的结论表明, 当 A 对称正定时, “最佳”方式 (2) 和 (3) 是等价的, 因此条件 (3) 对应的算法是 CG 算法.

定理 6.24 设 A 对称正定, 则

$$x^{(m)} = \arg \min_{x \in x^{(0)} + \mathcal{K}_m} \|x - x_*\|_A \tag{6.41}$$

当且仅当

$$x^{(m)} \in x^{(0)} + \mathcal{K}_m \quad \text{且} \quad b - Ax^{(m)} \perp \mathcal{K}_m. \tag{6.42}$$

证明. 首先证明充分性. 设 $x^{(m)}$ 满足 (6.42). 记 $\tilde{x} = x^{(m)} - x^{(0)}$, 则 $\tilde{x} \in \mathcal{K}_m$ 且

$$r_0 - A\tilde{x} \perp \mathcal{K}_m.$$

由正交投影的性质 (1.17) 可知, $\tilde{x}$ 是最佳逼近问题

$$\min_{x \in \mathcal{K}_m} \|x - A^{-1}r_0\|_A$$

的解, 即

$$\begin{aligned}\tilde{x} &= \arg \min_{x \in \mathcal{K}_m} \|x - A^{-1}r^{(0)}\|_A \\ &= \arg \min_{x \in \mathcal{K}_m} \|x - A^{-1}(b - Ax^{(0)})\|_A \\ &= \arg \min_{x \in \mathcal{K}_m} \|(x^{(0)} + x) - x_*\|_A.\end{aligned}$$

所以, $x^{(m)} = x^{(0)} + \tilde{x}$ 是最佳逼近问题

$$\min_{x \in x^{(0)} + \mathcal{K}_m} \|x - x_*\|_A$$

的解, 即结论 (6.41) 成立.

必要性只需利用正交投影的性质 (1.17) 即可, 见作业 6.13. □

6.8.3 GMRES 迭代算法

GMRES 算法是目前求解非对称线性方程组的最常用算法之一. 在该算法中, "最佳" 近似解的判别方法为 "使得 $\|r_m\|_2 = \|b - Ax^{(m)}\|_2$ 最小". 此时, "最佳" 近似解 $x^{(m)}$ 满足下面的性质.

定理 6.25 设 $A \in \mathbb{R}^{n\times n}$, 则

$$x^{(m)} = \arg \min_{x\in x^{(0)}+\mathcal{K}_m} \|b - Ax\|_2 \tag{6.43}$$

当且仅当

$$x^{(m)} \in x^{(0)} + \mathcal{K}_m \quad 且 \quad b - Ax^{(m)} \perp A\mathcal{K}_m, \tag{6.44}$$

其中 $A\mathcal{K}_m = \mathrm{span}\{Ar, A^2r, \ldots, A^mr\}$.

证明. 设 $x \in x^{(0)} + \mathcal{K}_m$, 则 x 可写为 $x = x^{(0)} + w$, 其中 $w \in \mathcal{K}_m$. 所以问题 (6.43) 可改写为

$$\min_{w\in\mathcal{K}_m} \|b - A(x^{(0)} + w)\|_2 = \min_{w\in\mathcal{K}_m} \|r_0 - Aw\|_2 \tag{6.45}$$

$$= \min_{z\in A\mathcal{K}_m} \|r_0 - z\|_2. \tag{6.46}$$

由正交投影的性质 (1.17) 可知, $\tilde{z}$ 是 (6.45) 的解当且仅当

$$r_0 - \tilde{z} \perp A\mathcal{K}_m.$$

充分性. 设 $x^{(m)}$ 满足 (6.44), 令 $w^{(m)} = x^{(m)} - x^{(0)}$. 则 $r_0 - Aw^{(m)} = b - Ax^{(m)}$, 所以

$$r_0 - Aw^{(m)} \perp A\mathcal{K}_m,$$

即 $\tilde{z} = Aw^{(m)}$ 是问题 (6.46) 的解, 也即 $w^{(m)}$ 是问题 (6.45) 的解. 因此, $x^{(m)}$ 是问题 (6.43) 的解.

必要性. 设 $x^{(m)}$ 是问题 (6.43) 的解. 令 $w^{(m)} = x^{(m)} - x^{(0)}$, 则 $\tilde{z} = Aw^{(m)}$ 是问题 (6.46) 的解. 所以 $r_0 - Aw^{(m)} \perp A\mathcal{K}_m$. 又 $r_0 - Aw^{(m)} = b - Ax^{(m)}$, 所以

$$b - Ax^{(m)} \perp A\mathcal{K}_m.$$

□

下面我们根据最优性条件 (6.43) 导出 GMRES 算法.

设迭代初始向量为 $x^{(0)}$, 则对任意向量 $x \in x^{(0)} + \mathcal{K}_m$, 可设 $x = x^{(0)} + V_my$, 其中 $y \in \mathbb{R}^m$. 于是有

$$\begin{aligned} r &= b - Ax \\ &= b - A(x^{(0)} - V_my) \\ &= r_0 - AV_my \end{aligned}$$

$$
\begin{aligned}
&= \beta v_1 - V_{m+1} H_{m+1,m} y \\
&= V_{m+1} \left(\beta e_1 - H_{m+1,m} y \right),
\end{aligned}
$$

这里 $\beta = \|r_0\|_2$. 由于 V_{m+1} 列正交, 所以

$$
\|r\|_2 = \|V_{m+1}(\beta e_1 - H_{m+1,m} y)\|_2 = \|\beta e_1 - H_{m+1,m} y\|_2.
$$

于是最优性条件 (6.43) 就转化为

$$
y^{(m)} = \arg \min_{y \in \mathbb{R}^m} \|\beta e_1 - H_{m+1,m} y\|_2. \tag{6.47}
$$

这是一个最小二乘问题. 由于 $H_{m+1,m}$ 是一个上 Hessenberg 矩阵, 且通常 m 不是很大, 所以我们可以用基于 Givens 变换的 QR 分解来求解. 下面就是 GMRES 算法的一个基本框架.

算法 6.13 GMRES 迭代算法基本框架

1: 给定初值 $x^{(0)}$, 停机标准 $\varepsilon > 0$, 以及最大迭代步数 IterMax
2: $r_0 = b - Ax^{(0)}$, $\beta = \|r_0\|_2$
3: $v_1 = r_0 / \beta$
4: **for** $j = 1$ to IterMax **do**
5:     $w = Av_j$
6:     **for** $i = 1$ to j **do**   % Arnoldi 过程
7:         $h_{i,j} = (v_i, w)$
8:         $w = w - h_{i,j} v_i$
9:     **end for**
10:     $h_{j+1,j} = \|w\|_2$
11:     **if** $h_{j+1,j} = 0$ **then**
12:         $m = j$, break
13:     **end if**
14:     $v_{j+1} = w / h_{j+1,j}$
15:     relres = $\|r_j\|_2 / \beta$ % 相对残量
16:     **if** relres$< \varepsilon$ **then**
17:         $m = j$, break
18:     **end if**
19: **end for**
20: 解最小二乘问题 (6.47), 得到 $y^{(m)}$
21: $x^{(m)} = x^{(0)} + V_m y^{(m)}$

具体实施细节

需要解决的问题有:

(1) 如何计算残量 $r_m \triangleq b - Ax^{(m)}$ 的范数?

(2) 如何求解最小二乘问题 (6.47)?

这两个问题可以同时处理. 由于 m 通常比较小, 因此我们采用 QR 分解来求解最小二乘问题. 设 $H_{m+1,m}$ 的 QR 分解为

$$H_{m+1,m} = Q_{m+1}^T R_{m+1,m},$$

其中 $Q_{m+1} \in \mathbb{R}^{(m+1)\times(m+1)}$ 是正交矩阵, $R_{m+1,m} \in \mathbb{R}^{(m+1)\times m}$ 是上三角矩阵. 则

$$\|\beta e_1 - H_{m+1,m}y\|_2 = \|\beta Q_{m+1}e_1 - R_{m+1,m}y\|_2 = \left\| \beta q_1 - \begin{bmatrix} R_m \\ 0 \end{bmatrix} y \right\|_2,$$

其中 $R_m \in \mathbb{R}^{m\times m}$ 是非奇异上三角矩阵 (这里假定 $H_{m+1,m}$ 不可约). 所以问题 (6.47) 的解为

$$y^{(m)} = \beta R_m^{-1} q_1(1:m),$$

且

$$\|r_m\|_2 = \|b - Ax^{(m)}\|_2 = \|\beta e_1 - H_{m+1,m}y^{(m)}\|_2 = \beta \cdot |q_1(m+1)|,$$

其中 $q_1(m+1)$ 表示 q_1 的第 $m+1$ 个分量.

$H_{m+1,m}$ 的 QR 分解的递推计算方法

由于 $H_{m+1,m}$ 是上 Hessenberg 矩阵, 因此我们采用 Givens 变换.

(1) 当 $m=1$ 时, $H_{21} = \begin{bmatrix} h_{11} \\ h_{21} \end{bmatrix}$, 构造 Givens 变换 G_1 使得

$$G_1 H_{21} = \begin{bmatrix} * \\ 0 \end{bmatrix} = R_{21}, \quad \text{即} \quad H_{21} = G_1^T R_{21}.$$

(2) 假定存在 $G_1, G_2, \dots, G_{m-1}$, 使得

$$(G_{m-1}\cdots G_2 G_1) H_{m,m-1} = R_{m,m-1},$$

即

$$H_{m,m-1} = (G_{m-1}\cdots G_2 G_1)^T R_{m,m-1} \triangleq Q_m^T R_{m,m-1}.$$

☞ 为了书写方便, 这里假定 G_i 的维数自动扩张, 以满足矩阵乘积的需要.

(3) 考虑 $H_{m+1,m}$ 的 QR 分解. 易知

$$H_{m+1,m} = \begin{bmatrix} H_{m,m-1} & h_m \\ 0 & h_{m+1,m} \end{bmatrix}, \quad \text{其中} \quad h_m = [h_{1m}, h_{2m}, \dots, h_{mm}]^T.$$

所以有

$$\begin{bmatrix} Q_m & 0 \\ 0 & 1 \end{bmatrix} H_{m+1,m} = \begin{bmatrix} R_{m,m-1} & Q_m h_m \\ 0 & h_{m+1,m} \end{bmatrix} = \begin{bmatrix} R_{m-1} & \tilde{h}_{m-1} \\ 0 & \hat{h}_{mm} \\ 0 & h_{m+1,m} \end{bmatrix},$$

其中 $\tilde{h}_{m-1}$ 是 $Q_m h_m$ 的前 $m-1$ 个元素组成的向量, $\hat{h}_{mm}$ 是 $Q_m h_m$ 的最后一个元素. 构造 Givens 变换 G_m:

$$G_m = \begin{bmatrix} I_{m-1} & 0 & 0 \\ 0 & c_m & s_m \\ 0 & -s_m & c_m \end{bmatrix} \in \mathbb{R}^{(m+1)\times(m+1)},$$

其中

$$c_m = \frac{\hat{h}_{m,m}}{\tilde{h}_{m,m}}, \quad s_m = \frac{h_{m+1,m}}{\tilde{h}_{m,m}}, \quad \tilde{h}_{m,m} = \sqrt{\hat{h}_{m,m}^2 + h_{m+1,m}^2}.$$

令

$$Q_{m+1} = G_m \begin{bmatrix} Q_m & 0 \\ 0 & 1 \end{bmatrix},$$

则

$$Q_{m+1}H_{m+1,m} = G_m \begin{bmatrix} R_{m-1} & \tilde{h}_{m-1} \\ 0 & \hat{h}_{j,j} \\ 0 & h_{m+1,m} \end{bmatrix} = \begin{bmatrix} R_{m-1} & \tilde{h}_{m-1} \\ 0 & \tilde{h}_{j,j} \\ 0 & 0 \end{bmatrix} \triangleq R_{m+1,m}.$$

所以可得 $H_{m+1,m}$ 的 QR 分解 $H_{m+1,m} = Q_{m+1}^T R_{m+1,m}$.

☞ 由 $H_{m,m-1}$ 的 QR 分解到 $H_{m+1,m}$ 的 QR 分解, 我们需要

(1) 计算 $Q_m h_m$, 即将之前的 $m-1$ 个 Givens 变换作用到 $H_{m+1,m}$ 的最后一列的前 m 个元素上, 所以我们需要保留所有的 Givens 变换;

(2) 残量计算: $\|r_m\|_2 = |\beta q_1(m+1)| = |\beta Q_{m+1}(m+1,1)|$, 即 $G_m G_{m-1} \cdots G_2 G_1(\beta e_1)$ 的最后一个分量的绝对值. 由于在计算 r_{m-1} 时就已经计算 $G_{m-1} \cdots G_2 G_1(\beta e_1)$, 因此这里只需一次 Givens 变换即可;

(3) $y^{(m)}$ 的计算: 当相对残量满足精度要求时, 需要计算 $y^{(m)} = R_m^{-1} q_1(1:m)$, 而 q_1 即为 $G_m G_{m-1} \cdots G_2 G_1(\beta e_1)$.

算法 6.14 实用 GMRES 算法

```
1: 给定初值 x^(0), 停机标准 ε > 0, 以及最大迭代步数 IterMax
2: r_0 = b − Ax^(0), β = ‖r_0‖_2
3: if β < ε then
4:     停止计算, 输出近似解 x^(0)
5: end if
6: v_1 = r_0/β
7: ξ = βe_1      % 记录 q_1
8: for j = 1 to IterMax do
9:     w = Av_j
10:    for i = 1 to j do      % Arnoldi 过程
11:        h_{i,j} = (v_i, w)
12:        w = w − h_{i,j} v_i
13:    end for
14:    h_{j+1,j} = ‖w‖_2
15:    if h_{j+1,j} = 0 then      % 迭代中断
16:        m = j, break
17:    end if
18:    v_{j+1} = w/h_{j+1,j}
19:    for i = 1 to j − 1 do      % 计算 G_{j−1} ··· G_2 G_1 H_{j+1,j}(1:j, j)
```

20: $\begin{bmatrix} h_{ij} \\ h_{i+1,j} \end{bmatrix} = \begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix} \begin{bmatrix} h_{ij} \\ h_{i+1,j} \end{bmatrix}$
21: **end for**
22: **if** $|h_{jj}| > |h_{j+1,j}|$ **then** % 构造 Givens 变换 G_j
23: $\tau = h_{j+1,j}/h_{jj}, c_j = 1/\sqrt{1+\tau^2}, s_j = c_j\tau$
24: **else**
25: $\tau = h_{jj}/h_{j+1,j}, s_j = 1/\sqrt{1+\tau^2}, c_j = s_j\tau$
26: **end if**
27: $h_{jj} = c_j h_{jj} + s_j h_{j+1,j}, h_{j+1,j} = 0$ % 计算 $G_j H_{j+1,j}(1:j,j)$
28: $\begin{bmatrix} \xi_j \\ \xi_{j+1} \end{bmatrix} = \begin{bmatrix} c_j & s_j \\ -s_j & c_j \end{bmatrix} \begin{bmatrix} \xi_{jj} \\ \xi_{j+1,j} \end{bmatrix}$ % 计算 $G_j(\beta G_{j-1} \cdots G_2 G_1 e_1)$
29: relres = $\|\xi_{j+1}\|_2/\beta$ % 相对残量
30: **if** relres$< \varepsilon$ **then**
31: $m = j$, break
32: **end if**
33: **end for**
34: $m = j$
35: $y^{(m)} = H(1:m, 1:m)\backslash\xi(1:m)$ % 求最小二乘问题的解, 回代求解
36: $x^{(m)} = x^{(0)} + V_m y^{(m)}$
37: **if** relres$< \varepsilon$ **then**
38: 输出近似解 x 及相关信息
39: **else**
40: 输出算法失败信息
41: **end if**

GMRES 算法的中断

在上面的 GMRES 算法中, 当执行到某一步时有 $h_{j+1,j} = 0$, 则算法会中断 (breakdown). 如果出现这种中断, 则我们就可以找到精确解.

定理 6.26 设 $A \in \mathbb{R}^{n\times n}$ 非奇异且 $r_0 \neq 0$. 若 $h_{i+1,i} \neq 0, i = 1, 2, \ldots, k-1$, 则 $h_{k+1,k} = 0$ 当且仅当 $x^{(k)}$ 是方程的精确解. (不考虑舍入误差)

证明. 必要性. 设 $h_{k+1,k} = 0$, 则有

$$AV_k = V_k H_k, \quad y^{(k)} = H_k^{-1}(\beta e_1).$$

所以

$$\|r_k\|_2 = \|b - Ax^{(k)}\|_2 = \|b - A(x^{(0)} + V_k y^{(k)})\|_2 = \|r_0 - V_k H_k y^{(k)}\|_2 = \|\beta v_1 - V_k(\beta e_1)\|_2 = 0.$$

充分性. 设 $x^{(k)}$ 是精确解, 则

$$0 = b - Ax^{(k)} = r_0 - V_{k+1} H_{k+1,k} y^{(k)} = V_{k+1}(\beta e_1 - H_{k+1,k} y^{(k)}).$$

反证法, 假设 $h_{k+1,k} \neq 0$, 则 $v_{k+1} \neq 0$. 因此 V_{k+1} 单位列正交, 故列满秩, 所以由上式可知

$$\beta e_1 - H_{k+1,k} y^{(k)} = 0.$$

由于 $H_{k+1,k}$ 是上 Hessenberg 矩阵, 且 $h_{i+1,i} \neq 0, i = 1, 2, \ldots, k$. 通过向后回代求解可得 $y^{(k)} = 0$, 于是 $\beta = 0$. 这与 $r_0 \neq 0$ 矛盾. 所以 $h_{k+1,k} = 0$ □

带重启的 GMRES 算法

由于随着迭代步数的增加, GMRES 算法的每一步所需的运算量和存储空间会越来越大, 因此当迭代步数很大时, GMRES 算法就不太实用. 通常的解决方法就是重启, 即事先设定一个重启迭代步数 k, 如 $k = 20$ 或 50 等等, 当 GMRES 达到这个迭代步数时仍不收敛, 则计算出方程组在 $x^{(0)} + \mathcal{K}_k$ 中的最佳近似解 $x^{(k)}$, 然后令 $x^{(0)} = x^{(k)}$, 并重新开始新的 GMRES 迭代. 不断重复该过程, 直到收敛为止.

算法 6.15 重启 GMRES 算法 (GMRES(k))

1: 设定重启步数 k ($\ll n$)
2: 给定初值 $x^{(0)}$, 停机标准 $\varepsilon > 0$, 以及最大迭代步数 IterMax
3: $r_0 = b - Ax^{(0)}$, $\beta = \|r_0\|_2$
4: **if** $\beta < \varepsilon$ **then**
5:     停止计算, 输出近似解 $x = x^{(0)}$
6: **end if**
7: $v_1 = r_0/\beta$
8: $\xi = \beta e_1$
9: **for** iter=1 to `ceil`(IterMax$/k$) **do**   % 外循环
10:     **for** $j = 1$ to k **do**
11:         调用 GMRES 循环
12:     **end for**
13:     $m = j$
14:     $y^{(m)} = H(1:m, 1:m) \backslash \xi(1:m)$
15:     $x^{(m)} = x^{(0)} + V_m y^{(m)}$
16:     **if** relres$< \varepsilon$ **then**
17:         break
18:     **end if**
19:     $x^{(0)} = x^{(m)}$ % 重启 GMRES
20:     $r_0 = b - Ax^{(0)}$, $\beta = \|r_0\|_2$
21:     $v_1 = r_0/\beta$
22:     $\xi = \beta e_1$
23: **end for**
24: **if** relres$< \varepsilon$ **then**
25:     输出近似解 $x^{(m)}$ 及相关信息
26: **else**
27:     输出算法失败信息

28: **end if**

带重启的 GMRES 算法需要注意的问题:

(1) 如何选取合适的重启步数 k? 一般只能依靠经验来选取;

(2) 不带重启的 GMRES 算法能保证算法的收敛性, 但带重启的 GMRES 算法却无法保证, 有时可能出现停滞现象 (*stagnation*).

6.8.4 共轭梯度法 (CG)

当 A 对称正定时, Arnoldi 过程就转化为 Lanczos 过程, 且

$$\begin{aligned} AV_m &= V_{m+1}T_{m+1,m} = V_mT_m + \beta_m v_{m+1}e_m^T, \\ V_m^T AV_m &= T_m, \end{aligned}$$

其中 $T_m = \mathrm{tridiag}(\beta_i, \alpha_{i+1}, \beta_{i+1})$, 见 (6.38). 由定理 6.24 可知, 此时我们需要在 $x^{(0)} + \mathcal{K}_m$ 寻找最优解 $x^{(m)}$, 满足

$$b - Ax^{(m)} \perp \mathcal{K}_m. \tag{6.48}$$

设 A 对称正定, 下面就根据这个性质推导 CG 算法的迭代公式.

首先, 设 $x^{(m)} = x^{(0)} + V_m z^{(m)}$, 其中 $z^{(m)} \in \mathbb{R}^m$. 由 (6.48) 可知

$$0 = V_m^T(b - Ax^{(m)}) = V_m^T(r_0 - AV_m z^{(m)}) = V_m^T(\beta v_1) - V_m^T AV_m z^{(m)} = \beta e_1 - T_m z^{(m)},$$

因此,

$$z^{(m)} = T_m^{-1}(\beta e_1).$$

由于 T_m 对称正定, 所以 T_m 存在分解 $T_m = L_m D_m L_m^T$. 于是可得

$$x^{(m)} = x^{(0)} + V_m z^{(m)} = x^{(0)} + V_m T_m^{-1}(\beta e_1) = x^{(0)} + (V_m L_m^{-T})(\beta D_m^{-1} L_m^{-1} e_1).$$

如果 $x^{(m)}$ 满足精度要求, 则计算结束. 否则我们需要计算

$$x^{(m+1)} = x^{(0)} + V_{m+1}T_{m+1}^{-1}(\beta e_1) = x^{(0)} + (V_{m+1}L_{m+1}^{-T})(\beta D_{m+1}^{-1} L_{m+1}^{-1} e_1).$$

下面就考虑如何利用递推方式来计算 $x^{(m)}, m = 1, 2, \ldots$. 记

$$\begin{aligned} \tilde{P}_m &= V_m L_m^{-T} = [\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_m] \in \mathbb{R}^{n\times m}, \\ y_m &= \beta D_m^{-1} L_m^{-1} e_1 = [\eta_1, \ldots, \eta_m]^T \in \mathbb{R}^m. \end{aligned}$$

我们首先证明下面的结论.

引理 6.10 下面的递推公式成立:

$$\begin{aligned} \tilde{P}_{m+1} &\triangleq V_{m+1}L_{m+1}^{-T} = [\tilde{P}_m, \tilde{p}_{m+1}] \\ y_{m+1} &\triangleq \beta D_{m+1}^{-1} L_{m+1}^{-1} e_1 = [y_m^T, \eta_{m+1}]^T. \end{aligned}$$

证明. 设 T_m 的 Cholesky 分解为

$$T_m = L_m D_m L_m^T = \begin{bmatrix} 1 & & & \\ l_1 & 1 & & \\ & \ddots & \ddots & \\ & & l_{m-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_m \end{bmatrix} \begin{bmatrix} 1 & & & \\ l_1 & 1 & & \\ & \ddots & \ddots & \\ & & l_{m-1} & 1 \end{bmatrix}^T.$$

有待定系数法可知 $d_1 = \alpha_1$,

$$l_i = \beta_i / d_i, \quad d_{i+1} = \alpha_{i+1} - l_i \beta_i, \quad i = 1, 2, \dots, m-1.$$

记 $\gamma = [0, \dots, 0, \beta_m]^T \in \mathbb{R}^m$ 则

$$\begin{aligned} T_{m+1} &= \begin{bmatrix} T_m & \gamma_m \\ \gamma_m^T & \alpha_{m+1} \end{bmatrix} = L_{m+1} D_{m+1} L_{m+1}^T \\ &= \begin{bmatrix} 1 & & & & \\ l_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & l_{m-1} & 1 & \\ & & & l_m & 1 \end{bmatrix} \begin{bmatrix} d_1 & & & & \\ & d_2 & & & \\ & & \ddots & & \\ & & & d_m & \\ & & & & d_{m+1} \end{bmatrix} \begin{bmatrix} 1 & & & & \\ l_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & l_{m-1} & 1 & \\ & & & l_m & 1 \end{bmatrix}^T, \end{aligned}$$

其中 $l_m = \beta_m / d_m$, $d_{m+1} = \alpha_{m+1} - l_m \beta_m$. 记 $\tilde{\gamma} = [0, \dots, 0, l_m]^T \in \mathbb{R}^m$, 则

$$L_{m+1} = \begin{bmatrix} L_m & 0 \\ \tilde{\gamma}^T & 1 \end{bmatrix} \quad \text{and} \quad L_{m+1}^{-1} = \begin{bmatrix} L_m^{-1} & 0 \\ -\tilde{\gamma}^T L_m^{-1} & 1 \end{bmatrix}.$$

所以有

$$\tilde{P}_{m+1} = V_{m+1} L_{m+1}^{-T} = [V_m, v_{m+1}] \begin{bmatrix} L_m^{-T} & -L_m^{-T} \tilde{\gamma} \\ 0 & 1 \end{bmatrix} = [V_m L_m^{-T}, -V_m L_m^{-T} \tilde{\gamma} + v_{m+1}].$$

又 $V_m L_m^{-T} \tilde{\gamma} = \tilde{P}_m [0, \dots, 0, l_m]^T = l_m \tilde{p}_m$, 所以 $\tilde{P}_{m+1} = [\tilde{P}_m, \tilde{p}_{m+1}]$, 其中

$$\boxed{\tilde{p}_{m+1} = -l_m \tilde{p}_m + v_{m+1}.} \tag{6.49}$$

另外,

$$\begin{aligned} y_{m+1} = \beta D_{m+1}^{-1} L_{m+1}^{-1} e_1 &= \beta \begin{bmatrix} D_m^{-1} & 0 \\ 0 & d_{m+1}^{-1} \end{bmatrix} \begin{bmatrix} L_m^{-1} & 0 \\ -\tilde{\gamma}^T L_m^{-1} & 1 \end{bmatrix} \begin{bmatrix} e_1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \beta D_m^{-1} L_m^{-1} e_1^m \\ -\beta \tilde{\gamma} L_m^{-1} e_1^m / d_{m+1} \end{bmatrix} \\ &= \begin{bmatrix} y_m \\ \eta_{m+1} \end{bmatrix}, \end{aligned}$$

即结论成立. □

有了上面的性质, 我们就可以得到从 $x^{(m)}$ 到 $x^{(m+1)}$ 的递推公式

$$\boxed{x_{m+1} = \tilde{P}_{m+1} y_{m+1} = [\tilde{P}_m, \tilde{p}_{m+1}] \begin{bmatrix} y_m \\ \eta_{m+1} \end{bmatrix} = x^{(m)} + \eta_{m+1} \tilde{p}_{m+1}.} \tag{6.50}$$

为了判断近似解是否满足要求, 我们需要计算残量 $r_m = b - Ax^{(m)}$. 易知, 残量满足下面的递推公式:

$$\boxed{r_{m+1} = b - Ax^{(m+1)} = b - A(x^{(m)} + \eta_{m+1}\tilde{p}_{m+1}) = r_m - \eta_{m+1} A\tilde{p}_{m+1}.} \tag{6.51}$$

另一方面, 我们有

$$\begin{aligned} r_m = b - Ax^{(m)} &= b - A(x^{(0)} + V_m z^{(m)}) \\ &= r_0 - AV_m z^{(m)} \\ &= \beta v_1 - V_m T_m z^{(m)} - \beta_m v_{m+1} e_m^T z^{(m)} \\ &= -\beta_m (e_m^T z^{(m)}) v_{m+1}, \end{aligned}$$

即 r_m 与 v_{m+1} 平行. 记

$$r_m = \tau_m v_{m+1}, \quad m = 0, 1, 2, \ldots,$$

其中

$$\tau_0 = \beta = \|r_0\|_2, \quad \tau_m = -\beta_m (e_m^T z^{(m)}), \quad m = 1, 2, \ldots.$$

记

$$\boxed{p_m = \tau_{m-1} \tilde{p}_m,}$$

于是我们就得到下面的递推公式

$$r_{m+1} = r_m - \eta_{m+1} A\tilde{p}_{m+1} = r_m - \xi_{m+1} A p_{m+1} \tag{6.52}$$

$$x^{(m+1)} = x^{(m)} + \eta_{m+1}\tilde{p}_{m+1} = x^{(m)} + \xi_{m+1} p_{m+1}, \tag{6.53}$$

其中 $\xi_{m+1} = \dfrac{\eta_{m+1}}{\tau_m}$, $m = 0, 1, 2, \ldots$, 以及

$$p_{m+1} = \tau_m \tilde{p}_{m+1} = \tau_m (v_{m+1} - l_m \tilde{p}_m) = r_m + \mu_m p_m, \tag{6.54}$$

其中 $\mu_m = -\dfrac{l_m \tau_m}{\tau_{m-1}}$, $m = 1, 2, \ldots$.

下面需要考虑系数 ξ_m 和 μ_m 的计算方法. 首先给出下面的性质.

引理 6.11 下面的结论成立:

(1) $r_1, r_2, \ldots, r_m$ 相互正交;

(2) $p_1, p_2, \ldots, p_m$ 相互 A-共轭 (或 A-正交), 即当 $i \neq j$ 时有 $p_i^T A p_j = 0$.

证明. (1) 由于 r_k 与 v_{k+1} 平行, 所以结论成立.

(2) 只需证明 $P_m^T A P_m$ 是对角矩阵即可, 即证 $\tilde{P}_m^T A \tilde{P}_m$ 是对角矩阵. 通过直接计算可得

$$\tilde{P}_m^T A \tilde{P}_m = (V_m L_m^{-T})^T A V_m L_m^{-T}$$

$$
\begin{aligned}
&= L_m^{-1} V_m^T A V_m L_m^{-T} \\
&= L_m^{-1} T_m L_m^{-T} \\
&= L_m^{-1} (L_m D_m L_m^T) L_m^{-T} \\
&= D_m.
\end{aligned}
$$

□

下面给出 ξ_m 和 μ_m 的递推公式. 在等式 (6.54) 两边同时左乘 $p_{m+1}^T A$ 可得

$$
p_{m+1}^T A p_{m+1} = p_{m+1}^T A r_m + \mu_m p_{m+1}^T A p_m = r_m^T A p_{m+1}.
$$

再用 r_m^T 左乘方程 (6.52) 可得

$$
0 = r_m^T r_{m+1} = r_m^T r_m - \xi_{m+1} r_m^T A p_{m+1},
$$

故

$$
\xi_{m+1} = \frac{r_m^T r_m}{r_m^T A p_{m+1}} = \frac{r_m^T r_m}{p_{m+1}^T A p_{m+1}}. \tag{6.55}
$$

等式 (6.54) 两边同时左乘 $p_m^T A$ 可得

$$
0 = p_m^T A p_{m+1} = p_m^T A r_m + \mu_m p_m^T A p_m,
$$

故

$$
\mu_m = -\frac{p_m^T A r_m}{p_m^T A p_m} = -\frac{r_m^T A p_m}{p_m^T A p_m}. \tag{6.56}
$$

为了进一步减少运算量, 我们将上式进行改写. 用 r_{m+1}^T 左乘方程 (6.52) 可得

$$
r_{m+1}^T r_{m+1} = r_{m+1}^T r_m - \xi_{m+1} r_{m+1} A p_{m+1} = -\xi_{m+1} r_{m+1} A p_{m+1},
$$

故

$$
\xi_{m+1} = -\frac{r_{m+1}^T r_{m+1}}{r_{m+1} A p_{m+1}}.
$$

所以 $\xi_m = -\dfrac{r_m^T r_m}{r_m^T A p_m}$, 即 $r_m^T A p_m = -r_m^T r_m / \xi_m$, 代入 (6.56) 可得

$$
\mu_m = -\frac{r_m^T A p_m}{p_m^T A p_m} = \frac{r_m^T r_m}{p_m^T A p_m} \cdot \frac{1}{\xi_m} = \frac{r_m^T r_m}{p_m^T A p_m} \cdot \frac{p_m^T A p_m}{r_{m-1}^T r_{m-1}} = \frac{r_m^T r_m}{r_{m-1}^T r_{m-1}}. \tag{6.57}
$$

又

$$
p_1 = \tau_0 \tilde{P}_1 = \beta v_1 = r_0,
$$

综合公式 (6.52), (6.53), (6.54) 和 (6.55), (6.57) 即可得 CG 算法:

算法 6.16 共轭梯度法 (CG)

1: 给定初值 $x_0 = 0$, 停机准则 $\varepsilon > 0$, 以及最大迭代步数 IterMax

2: $r_0 = b - A x_0$

3: $\beta = \|r_0\|_2$
4: **if** $\beta < \varepsilon$ **then**
5: 停止迭代, 输出近似解 $x^{(0)}$
6: **end if**
7: $p_1 = r_0$
8: **for** $m = 1$ to IterMax **do**
9: $\xi_m = (r_{m-1}^T r_{m-1})/(p_m^T A p_m)$
10: $x^{(m)} = x^{(m-1)} + \xi_m p_m$
11: $r_m = r_{m-1} - \xi_m A p_m$
12: relres $= \|r_m\|_2/\beta$
13: **if** relres$< \varepsilon$ **then**
14: break
15: **end if**
16: $\mu_m = (r_m^T r_m)/(r_{m-1}^T r_{m-1})$
17: $p_{m+1} = r_m + \mu_m p_m$
18: **end for**
19: **if** relres$< \varepsilon$ **then**
20: 输出近似解 $x^{(m)}$ 及相关信息
21: **else**
22: 输出算法失败信息
23: **end if**

☞ CG 算法的每个迭代步的主要运算为一个矩阵向量乘积和两个内积;

6.8.5 CG 算法的收敛性分析

设 x_* 是解析解, $x^{(m)} \in x^{(0)} + \mathcal{K}_m$ 是 CG 算法在 $x^{(0)} + \mathcal{K}_m$ 中找到的近似解, 即

$$x^{(m)} = \arg \min_{x \in x^{(0)} + \mathcal{K}_m} \|x - x_*\|_A.$$

记 $\mathbb{P}_k$ 为所有次数不超过 k 的多项式的集合. 对任意 $x \in x^{(0)} + \mathcal{K}_m$, 存在 $p(t) \in \mathbb{P}_{m-1}$, 使得

$$x = x^{(0)} + p(A) r_0.$$

于是有

$$x - x_* = \varepsilon_0 + p(A)(b - Ax^{(0)}) = \varepsilon_0 + p(A)(Ax_* - Ax^{(0)}) = (I - Ap(A))\varepsilon_0 \triangleq q(A)\varepsilon_0,$$

其中 $\varepsilon_0 = x^{(0)} - x_*$, 多项式 $q(t) = 1 - tp(t) \in \mathbb{P}_m$ 且 $q(0) = 1$. 所以

$$\|x - x_*\|_A^2 = \varepsilon_0^T q(A)^T A q(A) \varepsilon_0.$$

设 $A = Q\Lambda Q^T$ 是 A 的特征值分解, 其中 Q 是正交矩阵, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ 是对角阵, 且 $\lambda_i > 0$. 记 $y = [y_1, y_2, \ldots, y_n]^T \triangleq Q^T \varepsilon_0$, 则

$$\|x^{(m)} - x_*\|_A^2 = \min_{x \in x^{(0)} + \mathcal{K}_m} \|x - x_*\|_A^2$$

$$
\begin{aligned}
&= \min_{q\in\mathbb{P}_m,\, q(0)=1} \varepsilon_0^T q(A)^T A q(A) \varepsilon_0 \\
&= \min_{q\in\mathbb{P}_m,\, q(0)=1} \varepsilon_0^T Q q(\Lambda)^T \Lambda q(\Lambda) Q^T \varepsilon_0 \\
&= \min_{q\in\mathbb{P}_m,\, q(0)=1} y^T q(\Lambda)^T \Lambda q(\Lambda) y \\
&= \min_{q\in\mathbb{P}_m,\, q(0)=1} \sum_{i=1}^n y_i^2 \lambda_i q(\lambda_i)^2 \\
&\le \min_{q\in\mathbb{P}_m,\, q(0)=1} \max_{1\le i\le n} \{q(\lambda_i)^2\} \sum_{i=1}^n y_i^2 \lambda_i \\
&= \min_{q\in\mathbb{P}_m,\, q(0)=1} \max_{1\le i\le n} \{q(\lambda_i)^2\}\, y^T \Lambda y \\
&= \min_{q\in\mathbb{P}_m,\, q(0)=1} \max_{1\le i\le n} \{q(\lambda_i)^2\}\, \varepsilon_0^T A \varepsilon_0 \\
&= \min_{q\in\mathbb{P}_m,\, q(0)=1} \max_{1\le i\le n} \{q(\lambda_i)^2\}\, \|\varepsilon_0\|_A^2,
\end{aligned}
$$

即

$$
\frac{\|x^{(k)} - x_*\|_A^2}{\|x^{(0)} - x_*\|_A^2} \le \min_{q\in\mathbb{P}_m,\, q(0)=1} \max_{1\le i\le n} \{q(\lambda_i)^2\}.
$$

因此, 我们有下面的结论.

引理 6.12 *设 $A \in \mathbb{R}^{n\times n}$ 对称正定, $x^{(m)}$ 是 CG 算法迭代 m 步后得到的近似解. 则*

$$
\frac{\|x^{(m)} - x_*\|_A}{\|x^{(0)} - x_*\|_A} \le \min_{q\in\mathbb{P}_m,\, q(0)=1} \max_{1\le i\le n} |q(\lambda_i)|. \tag{6.58}
$$

由于 A 的特征值通常是不知道的, 因此不等式 (6.58) 右端通常难以计算. 但在许多情况下, 我们可以通过一些近似方法估计出 A 的最大特征值 λ_1 和最小特征值 λ_n. 假设 λ_1 和 λ_n 已知, 则不等式 (6.58) 右端可以放缩为

$$
\min_{q\in\mathbb{P}_m,\, q(0)=1} \max_{\lambda_n\le\lambda\le\lambda_1} |q(\lambda)|. \tag{6.59}
$$

由 Chebyshev 多项式的最佳逼近性质 (定理 6.21) 可知, 最小最大问题 (6.59) 的解为

$$
\tilde{q}(t) = \frac{T_m\left(\frac{2t-(\lambda_1+\lambda_n)}{\lambda_1-\lambda_n}\right)}{T_m\left(-\frac{\lambda_1+\lambda_n}{\lambda_1-\lambda_n}\right)},
$$

其中 T_m 为 m 次 Chebyshev 多项式. 由于 $T_m(t) = (-1)^m T_m(t)$, 且当 $|t| \le 1$ 时有 $|T_m(t)| \le 1$, 所以

$$
|\tilde{q}(t)| \le \frac{1}{T_m\left(\frac{\lambda_1+\lambda_n}{\lambda_1-\lambda_n}\right)}.
$$

又当 $|t| > 1$ 时

$$
T_m(t) = \frac{1}{2}\left[\left(t+\sqrt{t^2-1}\right)^m + \left(t+\sqrt{t^2-1}\right)^{-m}\right] \ge \frac{1}{2}\left(t+\sqrt{t^2-1}\right)^m,
$$

所以

$$
\begin{aligned}
T_m\left(\frac{\lambda_1+\lambda_n}{\lambda_1-\lambda_n}\right) &\geq \frac{1}{2}\left(\frac{\lambda_1+\lambda_n}{\lambda_1-\lambda_n}+\sqrt{\frac{(\lambda_1+\lambda_n)^2}{(\lambda_1-\lambda_n)^2}-1}\right)^m \\
&= \frac{1}{2}\left(\frac{\lambda_1+\lambda_n}{\lambda_1-\lambda_n}+\frac{2\sqrt{\lambda_1\lambda_n}}{\lambda_1-\lambda_n}\right)^m \\
&= \frac{1}{2}\left(\frac{\sqrt{\lambda_1}+\sqrt{\lambda_n}}{\sqrt{\lambda_1}-\sqrt{\lambda_n}}\right)^m \\
&= \frac{1}{2}\left(\frac{\sqrt{\kappa(A)}+1}{\sqrt{\kappa(A)}-1}\right)^m.
\end{aligned}
$$

其中 $\kappa(A)=\dfrac{\lambda_1}{\lambda_n}$ 为 A 的条件数. 因此我们可以得到下面的收敛性定理.

定理 6.27 设 $A\in\mathbb{R}^{n\times n}$ 对称正定, $x^{(m)}$ 是 CG 算法迭代 m 步后得到的近似解. 则

$$
\frac{\|x^{(m)}-x_*\|_A}{\|x^{(0)}-x_*\|_A} \leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^m. \tag{6.60}
$$

CG 算法的超收敛性

如果我们能够获得 A 的更多的特征值信息, 则能得到更好的误差限 [2].

定理 6.28 设 A 对称正定, 特征值为

$$
0<\lambda_n\leq\cdots\leq\lambda_{n+1-i}\leq b_1\leq\lambda_{n-i}\leq\cdots\leq\lambda_{j+1}\leq b_2\leq\lambda_j\leq\cdots\leq\lambda_1.
$$

则当 $m\geq i+j$ 时有

$$
\frac{\|x^{(m)}-x_*\|_A}{\|x^{(0)}-x_*\|_A} \leq 2\left(\frac{b-1}{b+1}\right)^{m-i-j} \max_{\lambda\in[b_1,b_2]}\left\{\prod_{k=n+1-i}^{n}\left(\frac{\lambda-\lambda_k}{\lambda_k}\right)\prod_{k=1}^{j}\left(\frac{\lambda_k-\lambda}{\lambda_k}\right)\right\}, \tag{6.61}
$$

其中

$$
b=\left(\frac{b_2}{b_1}\right)^{\frac{1}{2}}\geq 1.
$$

由此可知, 当 b_1 与 b_2 非常接近时, 迭代 $i+j$ 步后, CG 算法收敛会非常快.

推论 6.29 设 A 对称正定, 特征值为

$$
0<\delta\leq\lambda_n\leq\cdots\leq\lambda_{n+1-i}\leq 1-\varepsilon\leq\lambda_{n-i}\leq\cdots\leq\lambda_{j+1}\leq 1+\varepsilon\leq\lambda_j\leq\cdots\leq\lambda_1.
$$

则当 $m\geq i+j$ 时有

$$
\frac{\|x^{(m)}-x_*\|_A}{\|x^{(0)}-x_*\|_A} \leq 2\left(\frac{1+\varepsilon}{\delta}\right)^i \varepsilon^{m-i-j}. \tag{6.62}
$$

6.8.6 其它 Krylov 子空间迭代算法

Krylov 子空间算法一览表:

对称	CG (1952)	对称正定, 正交投影法 (Galerkin)
	MINRES (1975)	对称不定, 斜投影法 (Petrov-Galerkin)
	SYMMLQ (1975)	对称不定
	SQMR (1994)	对称不定
非对称	FOM (1981)	正交投影法, Arnoldi
	GMRES (1984)	斜投影法 (Petrov-Galerkin), Arnoldi
	BiCG (1976)	双正交 (biorthogonalization)
	QMR (1991)	双正交 (biorthogonalization)
	CGS (1989)	Transpose free
	BiCGStab (1992)	Transpose free, smoother convergence than CGS
	TFQMR (1993)	Transpose free, smoother convergence than CGS
	FGMRES (1993)	
正规方程	CGLS (1982)	最小二乘 (法方程)
	LSQR (1982)	最小二乘 (法方程)

6.9 预处理技术

见备课笔记（*To be continued ...* ）

6.10 课后习题

6.1 证明定理 6.7 (矩阵可约的充要条件):

设 $A \in \mathbb{R}^{n\times n}$, 指标集 $\mathbb{Z}_n = \{1, 2, \ldots, ..., n\}$. 则 A 可约的充要条件是存在非空指标集 $J \subset \mathbb{Z}_n$ 且 $J \neq \mathbb{Z}_n$, 使得

$$a_{ij} = 0, \quad i \in J \text{ 且 } j \in \mathbb{Z}_n \setminus J.$$

这里 $\mathbb{Z}_n \setminus J$ 表示 J 在 $\mathbb{Z}_n$ 中的补集.

6.2 已知 $A = \begin{bmatrix} a & 1 & 3 \\ 1 & a & 2 \\ -3 & 2 & a \end{bmatrix} \in \mathbb{R}^{3\times 3}$, 且 Jacobi 算法收敛, 求 α 的取值范围.

6.3 试给出红黑排序的模型问题的 SOR 算法的最优参数 ω_{opt}.

6.4 证明定理 6.12 (对称正定矩阵与古典迭代的收敛性):

设 $A \in \mathbb{R}^{n\times n}$ 对称正定.

(1) 若 $2D - A$ 正定, 则 Jacobi 迭代收敛;

(2) 若 $0 < \omega < 2$, 则 SOR 和 SSOR 收敛;

(3) G-S 迭代收敛.

6.5 证明定理 6.13 (古典迭代的收敛性与矩阵的对称正定性)

设 $A \in \mathbb{R}^{n\times n}$ 对称.

(1) 若 $2D - A$ 正定且 Jacobi 迭代收敛, 则 A 正定;

(2) 若 D 正定, 且存在 $\omega \in (0, 2)$ 使得 SOR (或 SSOR) 收敛, 则 A 正定;

(3) 若 D 正定, 且 G-S 迭代收敛, 则 A 正定.

6.6 设矩阵 $A \in \mathbb{C}^{n\times n}$ 是块三对角矩阵, 且主对角块为 0, 即

$$A = \begin{bmatrix} 0 & B_1 & & & \\ A_1 & 0 & B_2 & & \\ & A_2 & \ddots & \ddots & \\ & & \ddots & \ddots & B_{N-1} \\ & & & A_{N-1} & 0 \end{bmatrix}.$$

试证明: 矩阵 $A(\alpha) \triangleq \alpha L + \dfrac{1}{\alpha} U$ 的特征值与 α 无关 ($\alpha \in \mathbb{C}$ 且 $\alpha \neq 0$) , 其中 L 和 U 分别是 A 的下三角和上三角部分.

6.7 若存在非奇异矩阵 W, 使得 $W(I - G)W^{-1}$ 是对称正定矩阵, 则称迭代算法

$$x^{(k+1)} = Gx^{(k)} + g, \quad k = 0, 1, 2, \ldots$$

是可对称化的. 现假定上述迭代算法是可对称化的, 试证

(1) G 的特征值全部为实数, 且都小于 1;

(2) 存在 γ, 使得下面的外推迭代算法收敛

$$x^{(k+1)} = (1 - \gamma)x^{(k)} + \gamma(Gx^{(k)} + g), \quad k = 0, 1, 2, \ldots$$

6.8 设 $A = \begin{bmatrix} 2 & 0 \\ 2 & 1 \end{bmatrix}$, 构造迭代算法

$$x^{(k+1)} = x^{(k)} + \alpha(b - Ax^{(k)}), \quad k = 0, 1, 2, \ldots.$$

问: 当 $\alpha \in \mathbb{R}$ 取何值时, 该迭代算法收敛, 什么时候收敛最快?

6.9 设 $A \in \mathbb{R}^{n\times n}$.
证明: $\rho(A) < 1$ 的充要条件是 $I - A$ 非奇异, 且 $(I - A)^{-1}(I + A)$ 的特征值具有正实部.

6.10 设矩阵 A, B 都是实对称矩阵. 试证明: $AB = BA$ 的充要条件是存在正交矩阵 Q 使得

$$QAQ^T = \Lambda_A, \quad QBQ^T = \Lambda_B,$$

其中 Λ_A 和 Λ_B 分别表示由 A 和 B 的特征值构成的对角矩阵.

6.11 设 $A \in \mathbb{R}^{n\times n}$ 是对称三对角矩阵

$$A = \begin{bmatrix} a & b & & \\ b & \ddots & \ddots & \\ & \ddots & \ddots & b \\ & & b & a \end{bmatrix}.$$

计算矩阵 A 的特征值和特征向量.

6.12 设 $A \in \mathbb{R}^{n\times n}$ 对称正定, $B \in \mathbb{R}^{n\times k}$ 列满秩, 试证明 B^TAB 对称正定.

6.13 证明定理 6.24 中的必要性.

6.14 设 $p_1, p_2, \ldots, p_m$ 是 A-共轭的, 即当 $i \neq j$ 时有 $p_i^TAp_j = 0$, 试证明: $p_1, p_2, \ldots, p_m$ 线性无关.

思考题

6.15 设 $A \in \mathbb{R}^{n\times n}$. 证明: $\rho(A) < 1$ 的充要条件是存在对称正定矩阵 $P \in \mathbb{R}^{n\times n}$ 使得 $P - APA^T$ 正定.

第七章 特征值问题的迭代解法

当矩阵规模很大时, 计算其所有的特征值和特征向量是非常困难的. 而在实际应用中, 我们通常也只对其中的某些特征值和特征向量感兴趣, 因此也没必要计算所有的特征值和特征向量.

本章讨论计算部分特征值和特征值向量的迭代解法. 这些算法的存储量要远小于 $O(n^2)$, 运算量也远小于 $O(n^3)$.

参考资料

- B. N. Parlett, The Symmetric Eigenvalue Problem, Prentice Hall, Englewood Cliffs, NJ, 1980. (Republished by SIAM, Philadelphia, 1998.)
- Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia, 2000.
- Y. Saad, Numerical Methods for Large Eigenvalue Problems, 2nd revised edition. SIAM, Philadelphia, 2011.
- G. H. Golub and Ch. van Loan, Matrix Computations, 4th edition, Johns Hopkins University Press, Baltimore, 2013.
- Z. Bai, R. C. Li, and Y. F. Su, Lecture notes on Matrix Eigenvalue Computations, 2009.
- 课程: Prof. Bai, Large Scale Scientific Computing, 2013.
  http://www.cs.ucdavis.edu/~bai/ECS231/
- 课程: Prof. Dr. Peter Arbenz, Numerical Methods for Solving Large Scale Eigenvalue Problems, 2014.
  http://people.inf.ethz.ch/arbenz/ewp/
- 软件: ARPACK – collection of Fortran subroutines designed to compute a few eigenvalues and eigenvectors of large scale sparse matrices and pencils
  http://www.caam.rice.edu/software/ARPACK/

7.1 投影算法

最简单的特征值问题就是仅仅计算一个特征值, 如计算模最大的特征值. 这时我们可以使用幂迭代算法.

算法 7.1 幂迭代: 计算最大特征值

1: Given $x^{(0)}$
2: **for** $i = 1, 2, \ldots,$ until converge **do**
3: $\quad y^{(i)} = Ax^{(i-1)}$
4: $\quad x^{(i)} = y^{(i)}/\|y^{(i)}\|_2$
5: **end for**

幂迭代所产生的迭代向量 $x^{(0)}, x^{(1)}, \ldots, x^{(m-1)}$ 生成一个 Krylov 子空间

$$\mathcal{K}_m(A, x^{(0)}) = \text{span}\left\{x^{(0)}, Ax^{(0)}, \ldots, A^{m-1}x^{(0)}\right\}.$$

在幂迭代中, 我们取 $x^{(m-1)}$ 为近似特征向量. 显然, 如果我们在 $\mathcal{K}_m(A, x^{(0)})$ 中找出 "最佳" 的近似特征向量, 则收敛速度就可能会大大加快.

下面我们讨论如何在 $\mathcal{K}_m = \mathcal{K}_m(A, x^{(0)})$ 中寻找 "最佳" 的近似特征向量. 设 $A \in \mathbb{R}^{n\times n}$, 并设 $\mathcal{K}_m$ 和 $\mathcal{L}_m$ 是 $\mathbb{R}^n$ 的两个 m 维子空间. 投影算法就是在寻找 A 的近似特征对 $(\tilde{\lambda}, \tilde{x})$, 满足下面的 Petrov-Galerkin 条件

$$\text{find } \tilde{\lambda} \in \mathbb{C} \text{ and } \tilde{x} \in \mathcal{K}_m \quad \text{such that} \quad A\tilde{x} - \tilde{\lambda}\tilde{x} \perp \mathcal{L}_m. \tag{7.1}$$

这样的算法我们称为**斜投影算法**. 如果我们取 $\mathcal{L}_m = \mathcal{K}_m$, 则上面的算法就是一个**正交投影算法**, 此时条件 (7.1) 称为 Galerkin 条件.

设 $v_0, v_1, \ldots, v_{m-1}$ 和 $w_0, w_1, \ldots, w_{m-1}$ 分别是 $\mathcal{K}_m$ 和 $\mathcal{L}_m$ 的一组标准正交基. 令 $V_m = [v_0, v_1, \ldots, v_{m-1}]$, $W_m = [w_0, w_1, \ldots, w_{m-1}]$. 则对任意 $\tilde{x} \in \mathcal{K}_m$, 存在向量 $y \in \mathbb{R}^n$ 使得 $\tilde{x} = V_m y$, 即 $\tilde{x}$ 可以由 $v_0, v_1, \ldots, v_{m-1}$ 线性表出. 根据条件 (7.1), 我们可得

$$(AV_m y - \tilde{\lambda}V_m y, w_i) = 0, \quad i = 1, 2, \ldots, m,$$

即

$$W_m^T A V_m y = \tilde{\lambda} W_m^T V_m y. \tag{7.2}$$

这是一个广义特征值问题. 如果我们取 $\mathcal{L}_m = \mathcal{K}_m$, 并令 $W_m = V_m$, 则 (7.2) 就化为

$$T_m y = \tilde{\lambda} y, \tag{7.3}$$

其中 $T_m = V_m^T A V_m \in \mathbb{R}^{m\times m}$. 这意味着 $(\tilde{\lambda}, y)$ 是矩阵 T_m 的一个特征对. 由于 m 通常比较小, 因此我们可以使用先前讨论的方法 (如 QR 迭代) 来计算 $(\tilde{\lambda}, y)$. 这样我们就可以计算出 A 的一个近似特征对 $(\tilde{\lambda}, \tilde{x})$, 其中 $\tilde{x} = V_m y$.

7.2 Rayleigh-Ritz 算法

事实上, 我们可以在 $\mathcal{K}_m(A, x^{(0)})$ 中找出 m 个最佳近似特征向量及相应的最佳近似特征值. 这些近似特征值和近似特征向量就是 Ritz 值 和 Ritz 向量.

定义 7.1 设 $\mathcal{K}_m$ 是 $\mathbb{R}^{n\times n}$ 的一个 m 维子空间, 它的一组标准正交基为 $v_0, v_1, \ldots, v_{m-1}$, 并令 $V_m = [v_0, v_1, \ldots, v_{m-1}]$. 记 $T_m = V_m^T A V_m$, 设 $(\tilde{\lambda}, y)$ 是 T_m 的一组特征对, 即 $T_m y = \tilde{\lambda} y$ 且 $\|y\|_2 = 1$. 则我们成 $\tilde{\lambda}$ 是 A 的一个 Ritz 值, $\tilde{x} = V_m y$ 是 A 的一个 Ritz 向量.

Rayleigh-Ritz 算法 就是用 Ritz 值和 Ritz 向量来近似 A 的特征值与特征向量.

算法 7.2 Rayleigh Ritz procedure

1: Compute an orthonormal basis of $\mathcal{K}$: $V_m = [v_0, v_1, \ldots, v_{m-1}]$.
2: Compute the eigenvalues of the matrix $T_m = V_m^T A V_m$, *i.e.*, the Ritz values of A.
3: Select k desired ones: $\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_k$ where $k \le m$.
4: Compute the eigenvectors y_i of T_m associated with $\tilde{\lambda}_i$, $i = 1, 2, \ldots, k$.
5: Approximate the eigenvectors of A by the Ritz vectors $\tilde{x}_i = V_m y_i$, $i = 1, 2, \ldots, k$.

7.2.1 对称矩阵

这里我们讨论 A 对称时, Rayleigh Ritz 算法的性质.

设 $V_u \in \mathbb{R}^{n\times n-m}$ 是一个列正交矩阵, 且使得 $V = [V_m, V_u] \in \mathbb{R}^{n\times n}$ 是一个正交矩阵. 于是我们有

$$V^T A V = [V_m, V_u]^T A [V_m, V_u] = \begin{bmatrix} V_m^T A V_m & V_m^T A V_u \\ V_u^T A V_m & V_u^T A V_u \end{bmatrix}.$$

由于 A 对称的, 故 $T_m = V_m^T A V_m \in \mathbb{R}^{m\times m}$ 也是对称的. 此时, Ritz 值和 Ritz 向量具有下面的最优性质.

定理 7.1 设 $A \in \mathbb{R}^{n\times n}$ 对称, 则对任意的对称矩阵 $R \in \mathbb{R}^{m\times m}$, 有

$$\|AV_m - V_m R\|_2 \ge \|AV_m - V_m T_m\|_2,$$

即 $\|AV_m - V_m R\|_2$ 在 $R = T_m$ 处取最小值, 此时 $\|AV_m - V_m R\|_2 = \|V_u^T A V_m\|_2$.

证明. Let $R = T_m + Z$ where $Z \in \mathbb{R}^{m\times m}$ is symmetric. Note that both A and T_m are symmetric and $T_m = V_m^T A V_m$, we have

$$\begin{aligned} \|AV_m - V_m R\|_2^2 &= (AV_m - V_m R)^T (AV_m - V_m R) \\ &= \big(AV_m - V_m(T_m + Z)\big)^T \big(AV_m - V_m(T_m + Z)\big) \\ &= \|AV_m - V_m T_m\|_2^2 - V_m^T A V_m Z + T_m V_m^T V_m Z \\ &\quad - Z V_m^T A V_m + Z V_m^T V_m T_m + Z^T V_m^T V_m Z \\ &= \|AV_m - V_m T_m\|_2^2 + \|Z\|_2^2. \end{aligned}$$

It follows that $\|AV_m - V_m R\|_2^2$ is minimized when $Z = 0$ and

$$
\begin{aligned}
\|AV_m - V_m T_m\|_2 &= \left\|VV^T AV_m - V_m T_m\right\|_2 \\
&= \left\|(V_m, V_u)\begin{bmatrix} V_m^T AV_m \\ V_u^T AV_m \end{bmatrix} - V_m T_m\right\|_2 \\
&= \left\|V_u(V_u^T AV_m)\right\|_2 \\
&= \left\|V_u^T AV_m\right\|_2 .
\end{aligned}
$$

□

注: 定理 7.1 中的 2-范数可以改成任意的酉不变范数, 如 F-范数.

定理 7.2 设 $A \in \mathbb{R}^{n\times n}$ 对称, 并设 $T_m = U\Lambda U^T$ 是 $T_m = V_m^T AV_m$ 的特征值分解. 设 $Q \in \mathbb{R}^{n\times m}$ 是满足 $\mathrm{span}(Q) = \mathcal{K}$ 的任意单位列正交矩阵, $D \in \mathbb{R}^{m\times m}$ 是任意对角矩阵. 我们有

$$
\|AQ - QD\|_2 \geq \|AV_m - V_m T_m\|_2,
$$

且当 $Q = V_m U$, $D = \Lambda$ 时等式成立.

证明. 由于 $\mathrm{span}(Q) = \mathcal{K} = \mathrm{span}(V_m)$, 所以存在矩阵 $W \in \mathbb{R}^{m\times m}$ 使得 $Q = V_m W$. 又 Q 是单位列正交的, 因此

$$
I = Q^T Q = (V_m W)^T V_m W = W^T V_m^T V_m W = W^T W.
$$

这表明 W 是正交矩阵. 设 $WDW^T = T_m + Z$, 则

$$
\begin{aligned}
\|AQ - QD\|_2^2 &= \|AV_m W - V_m WD\|_2^2 \\
&= \|AV_m - V_m WDW^T\|_2^2 \\
&= \|AV_m - V_m T_m\|_2^2 + \|Z\|_2^2 \\
&\geq \|AV_m - V_m T_m\|_2^2.
\end{aligned}
$$

如果取 $W = U$ 和 $D = \Lambda$, 则 $Z = WDW^T - T_m = U\Lambda U^T - T_m = 0$. 此时上式中的等号成立. □

定理 7.2 表明, 在所有满足 $\mathrm{span}(Q) = \mathcal{K}$ 单位列正交矩阵 $Q \in \mathbb{R}^{n\times m}$ 和任意的对角矩阵 $D \in \mathbb{R}^m$ 中, 当 $Q = V_m U$ 和 $D = \Lambda$ 时, $\|AQ - QD\|_2$ 取到极小值.

定理 7.1 和定理 7.2 表明, 在 $\|AQ - QD\|_2$ 极小的意义下, Ritz 值是特征值的 “最佳” 近似. 所以我们用 Ritz 值作为特征值的近似是有道理的.

7.3 Lanczos 算法

设 $A \in \mathbb{R}^{n\times n}$ 是对称矩阵. Lanczos 就是利用 Lanczos 算法来计算 $\mathcal{K}_m$ 的基和 $T_m = V_m^T A V_m$, 然后计算 A 的 Ritz 值和 Ritz 向量.

算法 7.3 Lanczos Algorithm

1: Choose a vector v_0 such that $\|v_0\| = 1$, and set $\beta_0 = 0$
2: **for** $j = 0, 1, \ldots$ **do**
3: Compute $w = Av_j - \beta_j v_{j-1}$
4: $\alpha_{j+1} = (w, v_j)$
5: $w = w - \alpha_{j+1} v_j$
6: $\beta_{j+1} = \|w\|_2$
7: **if** $\beta_{j+1} = 0$ **then**
8: stop
9: **end if**
10: $v_{j+1} = w/\beta_{j+1}$
11: Compute the eigenvalues and eigenvectors of T_j
12: Check the convergence
13: **end for**

在 Lanczos 算法 7.3 中, 迭代 m 步后, 向量 $v_0, v_1, \ldots, v_{m-1}$ 构成子空间

$$\mathcal{K}_m(A, v_0) = \mathrm{span}\left\{v_0, Av_0, \ldots, A^{m-1}v_0\right\},$$

的一组基, 并且有

$$AV_m = V_m T_m + \beta_m v_m e_m^T,$$

其中 $e_m = [0, 0, \ldots, 0, 1]^T \in \mathbb{R}^m$,

$$T_m = V_m^T A V_m = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \beta_{m-1} & \alpha_m \end{bmatrix}.$$

设 $(\tilde{\lambda}, y)$ 是 T_m 的一个特征对, 则有

$$A(V_m y) = V_m T_m y + \beta_m v_m e_m^T y = \tilde{\lambda} V_m y + \beta_m (e_m^T y) v_m.$$

于是

$$Ax - \tilde{\lambda} x = \beta_m (e_m^T y) v_m,$$

即

$$\|Ax - \tilde{\lambda} x\|_2 = |\beta_m (e_m^T y)|,$$

其中 $x = V_m y$. 如果 $|\beta_m (e_m^T y)|$ 很小, 则我们就认为 $\tilde{\lambda}$ 是 A 的某个特征值的很好的近似. 事实上, 关于 Ritz 值 $\tilde{\lambda}$, 我们有下面的性质.

引理 7.1 设 $A \in \mathbb{R}^{n\times n}$ 是对称矩阵, 设 $r = Ax - \tilde{\lambda}x$, 其中 $x \neq 0$. 则

$$\min_{\lambda\in\sigma(A)} |\lambda - \tilde{\lambda}| \le \frac{\|r\|_2}{\|x\|_2},$$

其中 $\sigma(A)$ 表示 A 的谱, 即所有特征值组成的集合.

证明. 设 $A = U\Lambda U^T$ 是矩阵 A 的特征值分解. 则

$$r = (A - \tilde{\lambda}I)x = U(\Lambda - \tilde{\lambda}I)U^T x.$$

记 $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$. 对任意向量 $z = [z_1, z_2, \dots, z_n]^T \in \mathbb{R}^n$, 有

$$\begin{aligned}
\|(\Lambda - \tilde{\lambda}I)z\|_2^2 &= \sum_{i=1}^n (\lambda_i - \tilde{\lambda})^2 z_i^2 \\
&\ge \sum_{i=1}^n \min_{\lambda\in\sigma(A)} |\lambda - \tilde{\lambda}|^2 z_i^2 \\
&= \|z\|_2^2 \min_{\lambda\in\sigma(A)} |\lambda - \tilde{\lambda}|^2.
\end{aligned}$$

所以

$$\|r\|_2 = \|U^T r\|_2 = \|(\Lambda - \tilde{\lambda}I)U^T x\|_2 \ge \|U^T x\|_2 \min_{\lambda\in\sigma(A)} |\lambda - \tilde{\lambda}| = \|x\|_2 \cdot \min_{\lambda\in\sigma(A)} |\lambda - \tilde{\lambda}|.$$

□

由引理 7.1 可知, 存在 A 的某个特征值 λ, 使得

$$|\lambda - \tilde{\lambda}| \le \frac{\|\beta_m (e_m^T y) v_m\|_2}{\|V_m y\|_2} = \frac{|\beta_m| \cdot |e_m^T y|}{\|y\|_2} = |\beta_m| \cdot |e_m^T y|. \tag{7.4}$$

注: 在前面的讨论中, 我们都没有考虑实际计算时可能的误差. 在实际计算中, 由于浮点运算的误差, 即使 m 很小 (如 $m = 10$ 或 $m = 20$), 也可能会导致向量 $\{v_i\}$ 失去正交性. 这时我们必须采取一些补救措施, 最简单的方法就是对它们重新来一次正交话, 即在算法 7.3 的第 5 步后加上一条语句

$$w = w - \sum_{i=1}^j (w, v_i) v_i.$$

这个过程就称为带全正交过程的 Lanczos 算法. 显然, 这个过程是非常费时的.

另外一个可行的方法就是选择性正交.

7.4 Arnoldi 算法

这里考虑非对称情形, 即计算非对称矩阵 A 的特征值. 与 Lanczos 算法的思想相类似, 我们可以使用 Arnoldi 算法, 即通过 Arnoldi 算法计算 $\mathcal{K}_m$ 的标准正交基 $v_0, v_1, \ldots, v_{m-1}$ 和上 Hessenberg 矩阵 $H_m = V_m^T A V_m$, 使得

$$AV_m = V_m H_m + h_{m+1,m} v_m e_m^T \quad \text{and} \quad V_m^T A V_m = H_m.$$

但此时 H_m 只是上 Hessenberg, 而不是对称三对角. 但我们同样可以通过计算 H_m 的特征值和特征向量来得到 A 的 Ritz 值的 Ritz 向量, 并用它们来近似 A 的特征值和特征向量.

设 $(\tilde{\lambda}, y)$ 是 H_m 的一个特征对, 其中 $\|y\|_2 = 1$, 则

$$A(V_m y) = V_m H_m y + h_{m+1,m} v_m e_m^T y = \tilde{\lambda} V_m y + h_{m+1,m}(e_m^T y) v_m,$$

所以

$$\|Ax - \tilde{\lambda} x\|_2 = \|h_{m+1,m}(e_m^T y) v_m\|_2 = |h_{m+1,m}| \cdot |e_m^T y|,$$

其中 $x = V_m y$. 若 $|h_{m+1,m}| \cdot |e_m^T y|$ 足够小, 我就认为 $(\tilde{\lambda}, x)$ 是 A 的某个特征对的近似.

算法 7.4 Arnoldi Algorithm

1: Choose a vector v_0 such that $\|v_0\|_2 = 1$
2: **for** $j = 0, 1, \ldots$ **do**
3:     Compute $w_{j+1} = A v_j$
4:     **for** $i = 0, 1, \ldots, j$ **do**
5:         $h_{ij} = (w_{j+1}, v_i)$
6:         $w_{j+1} = w_{j+1} - h_{ij} v_i$
7:     **end for**
8:     $h_{j+1,j} = \|w_{j+1}\|_2$
9:     **if** $h_{j+1,j} = 0$ **then**
10:         stop
11:     **end if**
12:     $v_{j+1} = w_{j+1} / h_{j+1,j}$
13:     Compute the eigenvalues and eigenvectors of T_j
14:     Check the convergence
15: **end for**

由于 A 是非对称的, 其特征值可能是复的, 或者是坏条件的, 此时 Lanczos 算法的一些最优性质就不再成立. 尽快如此, 目前还是存在一些有效 Arnoldi 算法的实现方式, 可参见 [26, 34, 35].

7.5 非对称 Lanczos 算法

非对称 Lanczos 算法就是 Lanczos 算法在非对称矩阵上的推广, 它是基于 Lanczos 双正交化过程.

设 v_0 和 w_0 是任意的非零向量, 并设

$$\mathcal{K}_m(A, v_0) = \text{span}\{v_0, Av_0, \dots, A^{m-1}v_0\}$$

和

$$\mathcal{K}_m(A^T, w_0) = \text{span}\{w_0, A^T w_0, \dots, (A^T)^{m-1}w_0\}.$$

Lanczos 双正交化过程就是计算 $\mathcal{K}_m(A, v_0)$ 和 $\mathcal{K}_m(A, w_0)$ 的基 $\{v_i\}$ 和 $\{w_i\}$, 满足 $\{v_i\}$ 和 $\{w_i\}$ 相互正交, 即

$$(v_i, w_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}.$$

所以

$$W_m^T V_m = I,$$

其中 $V_m = [v_0, v_1, \dots, v_{m-1}]$, $W_m = [w_0, w_1, \dots, w_{m-1}]$.

注意, 通常 $\{v_i\}_{i=0}^m$ 或 $\{w_j\}_{j=0}^m$ 本身并不一定正交, 故 V_m 和 W_m 通常并不列正交.

根据 Lanczos 双正交化过程, 我们可得

$$\begin{aligned} AV_m &= V_m T_m + \gamma_m v_m e_m^T, \\ A^T W_m &= W_m T_m^T + \beta_m w_m e_m^T, \end{aligned}$$

其中 $e_m = [0, \dots, 0, 1]^T \in \mathbb{R}^m$,

$$T_m = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \gamma_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \gamma_{m-1} & \alpha_m \end{bmatrix}.$$

所以

$$W_m^T A V_m = W_m^T V_m T_m + \gamma_m (W_m^T v_m) e_m^T = T_m.$$

设 $\tilde{\lambda}$ 是 T_m 的特征值, 其对应的右特征向量和左特征向量分别为 y 和 z, 且 $\|y\|_2 = \|z\|_2 = 1$, 即

$$T_m y = \tilde{\lambda} y \quad \text{and} \quad z^T T_m = \tilde{\lambda} z^T.$$

于是有

$$\begin{aligned} &AV_m y = V_m T_m y + \gamma_m v_m e_m^T y = \tilde{\lambda} V_m y + \gamma_m e_m^T y v_m, \\ &(W_m z)^T A = (A^T W_m z)^T = (W_m T_m^T z)^T + \beta_m e_m^T z w_m^T = \tilde{\lambda}(W_m z)^T + \beta_m e_m^T z w_m^T. \end{aligned}$$

若 $|\gamma_m(e_m^T y)|$ 和 $|\beta_m(e_m^T z)|$ 足够小, 我们就认为 $\tilde{\lambda}$ 是 A 的某个特征值的近似, 而 $V_m y$ 和 $W_m z$ 就是相应的右特征向量和左特征向量的近似.

算法 7.5 Nonsymmetric Lanczos Algorithm

1: Choose two vectors v_0 and w_0 such that $(v_0, w_0) = 1$
2: Set $\beta_0 = 0$ and $\gamma_0 = 0$
3: **for** $j = 0, 1, \ldots$ **do**
4: Compute $\alpha_{j+1} = (Av_j, w_j)$
5: $\tilde{v}_{j+1} = Av_j - \alpha_{j+1}v_j - \beta_j v_{j-1}$
6: $\tilde{w}_{j+1} = A^T w_j - \alpha_{j+1}w_j - \gamma_j w_{j-1}$
7: $\gamma_{j+1} = |(\tilde{v}_{j+1}, \tilde{w}_{j+1})|^{1/2}$
8: **if** $\gamma_{j+1} = 0$ **then**
9: stop
10: **end if**
11: $\beta_{j+1} = (\tilde{v}_{j+1}, \tilde{w}_{j+1})/\gamma_{j+1}$
12: $v_{j+1} = \tilde{v}_{j+1}/\gamma_{j+1}$
13: $w_{j+1} = \tilde{w}_{j+1}/\beta_{j+1}$
14: Compute the eigenvalues and eigenvectors of T_j
15: Check the convergence
16: **end for**

非对称 Lanczos 算法的显著优点就是节省运算量, 缺点是更容易被中断.

EAST CHINA NORMAL UNIVERSITY

参考文献

[1] J. O. Aasen, On the reduction of a symmetric matrix to tridiagonal form, BIT, 11 (1971), 233–242.

[2] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, 1994.

[3] R. Barrett, et.al, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, 1994. (http://www.netlib.org/templates/index.html)

[4] Z.-Z. Bai, G. H. Golub and M. K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, *SIAM Journal on Matrix Analysis and Applications*, 24 (2003), 603–626.

[5] Åke Björck, Solving linear least square problems by Gram-Schmidt orthogonalization, BIT, 7 (1967), 1–21.

[6] Åke Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.

[7] J. R. Bunch and L. Kaufman, Some stable methods for calculating inertia and solving symmetric linear systems, Mathematics of Computation, 31 (1977), 163–179.

[8] J. J. M. Cuppen, A Divide and Conquer Method for the Symmetric Tridiagonal Eigenproblem, Numerische Mathematik, 36 (1981), 177–195.

[9] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.

[10] Z. Drmač and K. Veselić, New fast and accurate jacobi SVD algorithm. I SIAM Journal on Matrix Analysis and Applications, 29 (2008), 1322–1342.

[11] Z. Drmač and K. Veselić, New fast and accurate jacobi SVD algorithm. II SIAM Journal on Matrix Analysis and Applications, 29 (2008), 1343–1362.

[12] K. Fernando and B. Parlett, Accurate singular values and differential qd algorithms, Numerische Mathematik, 67 (1994), 191–229.

[13] N. Gastinel, *Linear Numerical Analysis*, Kershaw Publishing, London, 1083.

[14] G. H. Golub, History of numerical linear algebra: A personal view, Stanford, 2007. Available at http://forum.stanford.edu/events/2007slides/plenary/history-revised-2007-03-19-golub.pdf

[15] G. H. Golub and W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, SIAM Journal on Numerical Analysis, Series B, 2 (1965), 205–224.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The 4th Editon, The Johns Hopkins University Press, Baltimore, MD, 2013.

[17] M. Gu and S. C. Eisenstat, A stable algorithm for the rank-1 modification of the symmetric eigenproblem, SIAM Journal on Matrix Analysis and Applications, 15 (1994), 1266–1276.

[18] M. Gu and S. C. Eisenstat, A Divide-and-Conquer algorithm for the bidiagonal SVD, SIAM Journal on Matrix Analysis and Applications, 16 (1995), 79–92.

[19] M. Gu and S. C. Eisenstat, A Divide-and-Conquer algorithm for the symmetric tridiagonal eigenproblem, SIAM Journal on Matrix Analysis and Applications, 16 (1995), 172–191.

[20] A. Hadjidimos, Accelerated overrelaxation method, Mathematics of Computation, 32 (1978), 149–157.

[21] Nicholas J. Higham, *Accuracy and Stability of Numerical Algorithms*, Second Edition, SIAM, Philadelphia, 2002.

[22] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[23] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.

[24] W. Kahan Numerical Linear Algebra, Canadian Math. Bull., 9 (1966), 757–801.

[25] D. Kressner, *Numerical Methods for General and Structured Eigenvalue Problems*, Lecture Notes in Computational Sciences and Engineering 46, Springer-Verlag, 2005.

[26] R. Lehoucq, *Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration*, Ph.D. thesis, Rice University, Houston, TX, 1995.

[27] E. H. Moore, On the reciprocal of the general algebraic matrix, Bull. Amer. Math. Soc., 26 (1920), 394–395.

[28] Christopher C. Paige, Miroslav Rozložník and Zdeněk Strakoš, Modified Gram–Schmidt (MGS), least squares, and backward stability of MGS-GMRES, SIAM Journal on Matrix Analysis and Applications, (28) 2006, 264−-284.

[29] B. N. Parlett, *The Symmetric Eigenvalue Problem*, The 2nd Edition, SIAM, Philadelphia, PA, 1998.

[30] D. W. Peaceman and H. H. Rachford, Jr., The numerical solution of parabolic and elliptic differential equations, *Journal of the Society for Industrial and Applied Mathematics*, 3 (1955), 28–41.

[31] R. Penrose, A generalized inverse for matrices, Proc. Cambridge Philos. Soc., 51 (1955), 406–413.

[32] V. Britanak, P. Yip and K. Rao, *Discrete Cosine and Sine Transforms: General properties, Fast algorithms and Integer Approximations*, Academic Press, 2007.

[33] J. Rutter, *A Serial Implementation of Cuppen'S Divide and Conquer Algorithm for the Symmetric Eigenvalue Problem*, Master's Thesis, University of California, 1994.

[34] Y. Saad, *Numerical Methods for Large Eigenvalue Problems: Theory and Algorithms*, Manchester University Press, Manchester, UK, 1992.

[35] D. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, *SIAM Journal on Matrix Analysis and Applications*, 13 (1992), 357–385.

[36] G. W. Stewart, *Matrix Algorithms, Vol I: Basic Decomposition*, SIAM, Philadelphia, PA, 1998.

[37] G. W. Stewart, *Matrix Algorithms, Vol II: Eigensystems*, SIAM, Philadelphia, PA, 2001.

[38] G. W. Stewart and Ji-guang Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.

[39] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.

[40] L. N. Trefethen, Numerical Analysis, in *Princeton Companion to Mathematics*, Edited by T. Gowers, J. Barrow-Green and I. Leader, Princeton University Press, 2008.

[41] D. S. Watkins, *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*, SIAM, Philadelphia, 2007.

[42] D. S. Watkins and L. Elsner, Convergence of algorithms of decomposition type for the eigenvalue problem, *Linear Algebra and its Applications*, 143 (1991), 19–47.

[43] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford University, Oxford, 1965.

[44] R. S. Varga, *Matrix Iterative Analysis*, 2nd edition, Prentice-Hall, Englewood Cliffs, NJ, 2000. (1st edition 1962)

[45] D. M. Young, *Iterative Methods for Solving Partial Difference Equations of Elliptic Type*, PhD thesis, Harward University, 1950.

[46] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.

[47] 胡家赣, **线性代数方程组的迭代解法**, 科学出版社, 1991.

[48] 蒋尔雄, **矩阵计算**, 科学出版社, 2008.

[49] 李大明, **数值线性代数**, 清华大学出版社.

[50] 孙继广, **矩阵扰动分析**, 科学出版社, 北京, 2001.

[51] 魏木生, **广义最小二乘问题的理论与计算**, 科学出版社, 北京, 2006.

[52] 徐树方, **矩阵计算的理论与方法**, 北京大学出版社, 北京, 1995.

[53] 徐树方, 钱江, **矩阵计算六讲**, 高等教育出版社, 北京, 2011.